Università Roma Tre
'Dottorato in Filosofia e Teoria delle scienze umane'

joint doctoral degree with

Université Paris 13
'Ecole Doctorale Sciences, Technologies, Santé
specialité Informatique'

Ontologies, Logic and Interaction

Candidate: Marco Romano

April 26th, 2011

Supervisors:
V. Michele Abrusci
Christophe Fouqueré

Examination board:
V. Michele Abrusci (Università Roma Tre)
Bruno Bachimont (Univ. Technologie Compiègne)
Claudia Casadio (Univ. G. D'Annunzio, Chieti)
Christophe Fouqueré (Université Paris 13)
Adeline Nazarenko (Université Paris 13)
Remo Pareschi (Università del Molise)

Referees:
Bruno Bachimont
Andrzej Murawski (University of Leicester)

Ontologies, Logic and Interaction

Marco Romano

Abstract

Semantic Web is the ambitious project to enrich data in the Web with an "extra level of semantics" (basically the metadata) in order to enable intelligent web agents to use information in the Web, rather than only have it displayed on screen. Central in such a project is the role of ontologies as knowledge representations designed to signal to web agents what resources are and what they can be used for. This work aims to give some contribution to such a project by suggesting alternative perspectives from which to consider it. But also by proposing an original theoretical model, less constrained by the linguistic approach to symbolic representation, and most of all, capable to represent on a common logical ground both ontologies and folksonomies (the two faces of a common effort in Semantic Web and Web 2.0 to give structure to information). This model exploits a special class of Coherence spaces that we define as Ontological Compatibility Spaces (OCS). Together with OCSs, we propose also an interpretation of the query-answer interplay between web agents (and data sources), based on Ludics, that we deem capable to richly model the interactive processes by means of which information can be exchanged between systems that adopt different ontologies. Together, these proposals compose a rich and fascinating frame to consider, even from a purely philosophical point of view, issues like the process that leads to the definition of a concept in an ontology, and some major problems that emerge with the use of ontologies to enable communication, namely language translation issues and really ontological puzzles. The same issues, however, have much more general significance, since they hold in any case of free communication – that is not regimented in a rigid, artificial and predefined protocol. A recurring theme in this work is the tight and intricate relation that holds between ontology and the language. It is a sort of *leitmotiv* together with the themes of multiplicity – the dynamics of multiple, competing or collaborating ontologies – and of interaction, of which communication is just one form, the one that allows to observe precisely the relation between language and reality.

Contents

Introduction

A few words from one of the inventors of the World Wide Web will bring us quickly into the subject of our PhD thesis. They briefly express the colossal project to which we aspire to give some contribution with the results of our work – though not by proposing some technical enhancement for the ongoing activities, rather by suggesting alternative perspectives from which to consider the overall project. Tim Berners-Lee has introduced Semantic Web thus:

> To a computer, the Web is a flat, boring world, devoid of meaning. This is a pity, as in fact documents on the Web describe real objects and imaginary concepts, and give particular relationships between them. For example, a document might describe a person. The title document to a house describes a house and also the ownership relation with a person. Adding semantics to the Web involves two things: allowing documents which have information in machine-readable forms, and allowing links to be created with relationship values. Only when we have this extra level of semantics will we be able to use computer power to help us exploit the information to a greater extent than our own reading. [Berners-Lee, 1994]

About six years after these words dated 1994, the World Wide Web Consortium (W3C) started the *Semantic Web Initiative*, aiming precisely at making real that picture. The project to achieve Semantic Web, indeed, is inspired by this same idea of an "extra level of semantics". It is to be produced, according to the project of the W3C, by means of a series of superimposed layers of data intended to define, or at least describe, the meaning of data at the lower levels. Whereas the foundation of such a building is already in place, made of the same fundamental technologies of the World Wide Web, there is actually an intense activity on that which should prepare the very core of Semantic Web: the conceptual structuring of information.

As data in databases are meaningful inasmuch as they are framed in a logical structure, precisely the logical schema of the data base, so also data "sparse" in the Web need some logical frame to be(come) meaningful. W3C approached the issue in its own way, that is by defining standard languages such as XML, RDF and OWL. This last is intended to design *ontologies*, thus testifying the combined efforts of (part of) the Semantic Web community and most of the research in Knowledge Engineering (KE), that has taken up the challenge of Semantic Web maybe foreseeing in it the new environment where to begin another season for the (classical) Artificial Intelligence.

These *ontologies* indeed are among the richest forms of Knowledge Representation (KR) and also within the project for the Semantic Web have gained the leading role, accompanied in minor roles by other forms of knowledge representation, such as *rules* for instance – that by the way had been the favoured model during the season of *expert systems*. Ontologies therefore have been chosen to produce the top-level layer of description of data in the Web, so as to provide them with well defined semantics. Thanks to ontologies, then, software programs surfing the Web – like nowadays web agents, possibly somewhat "smarter" – should be able to find information (data plus some logical structure) and *use* it instead of only displaying it on our monitors.

Basically, the idea of the extra level of semantics can be summed up with the metaphor of labels. Semantic Web is a huge effort to stick labels to possibly every resource accessible through the World Wide Web, in such a way that computers may know what any given resource *is* by reading the metadata (data about data) on its label. Similarly, the labels stuck on every garment in a clothes shop are to give the customer the relevant information about any piece (what it is, what it is made of, its price and so on). Obviously, besides the labels stuck to the garments it needs also the inventory of what exists in the shop (in the Web), and this is where ontology comes into play, as a logically enriched form of catalogue. And finally it needs also that customers visiting the shop (the Web) can understand what is written on the labels. This point comprises not only language issues like the use of different terms (from different languages) to name the same thing (e.g. scarf instead of *écharpe* or *sciarpa*), but also ontological issues like the competence, the store of knowledge that allows a customer to buy a scarf if she feels cold on her neck. The ontology, as a logically enriched form of catalogue, should not only list what exists, but also express as much as possible of the relevant knowledge about the items existing in the world that it is to represent: the clothes shop as well as the Web. In this way knowledge is shared and the customer may look up the catalogue in order to discover what she misses about some term appearing on the label of some piece that she is interested in.

The metaphor of the labels and the example of the clothes shop have been already enough to highlight major problems with the use of ontologies, namely linguistic issues and really ontological puzzles. Although Semantic Web ontologies, as Knowledge Representation artefacts, are usually designed by knowledge engineers or computer scientists, with poor (or none) contribution from philosophy – especially whenever the KR ontology is to be used within some working application – we will insist, however, in signalling their philosophical significance as far as such ontologies are intended to support communication between agents that possibly do not speak the same language, that is face some of the problems that we have just alluded to.

A recurring theme in this thesis will be therefore the strict and intricate relation that holds between ontology and the language. It will be a sort of *leitmotiv* together with the themes of multiplicity – the dynamics of multiple, competing or collaborating ontologies – and of interaction, of which communication is just one form, the form that allows to observe precisely the relation between language and reality.

Our work indeed walks a path from the analysis of current efforts in order to enable machines (computers) to act in the Web on our behalf – giving them part of our knowledge about the world – to the speculation about our own knowledge of the world – precisely because it is to be passed to computers. It touches therefore our way of dealing with ontology, both in explicit settings and aware manners (e.g. when discussing on a philosophical basis about what is reality in our hypermediated world, but also when designing a domain ontology for use in a Web service); and implicitly whenever we act and communicate in the world.

While walking such a path we will focus on Knowledge Representation, as far as it is concerned with Semantic Web, and we will concentrate mainly on the conceptual and logico-mathematical assumptions that lye behind and underpin the theory and practice of KR [Baader et al., 2003, Staab and Studer, 2009]. We find that part of such assumptions are also well rooted in our minds (in particular the minds of people working within or "near" the Semantic Web initiative) conditioning our approach in describing and accessing information and knowledge [Smith, 1995, 2005]. We will raise then the question whether it is worthy to put in discussion such assumptions and make some room for other ideas, for an alternative approach, at least for use in the Web.

Our contribution will not be a further development in any particular aspect of the global project to have Semantic Web working. Rather we are interested in showing that some modifications about the standard way of thinking about how to formalize and use the information in the Web may open a field of new opportunities concerning how to achieve the next stage of World Wide Web.

Briefly, we aim at widening the possibilities of interaction between agents (and then increasing the usefulness of computers in accessing information over the Web) by proposing a theoretical simple framework capable to describe this kind of interaction (i.e. information exchanges) as the process occurring in a dialogical way between two agents which cooperate to accomplish some task. By the way, from a logical point of view, all this can be seen as a matter of building interactively a demonstration of some assertion.

Evolving Web

In the first part of our work we present the scenario of the World Wide Web and of its two lines of evolution: Semantic Web and Social Web. In particular we analyse the project proposed by the W3C for the construction of the Semantic Web and consider to which extent the cooptation of the branches of KE and KR may provide Semantic Web with the needed dose of logical structuring of information, so as to compose the missing layer of logic capable to unify the variety of data sources that Semantic Web should be made of.

We do not care how to properly build ontologies for the Web, simply because it is outside of our interest here. We then leave aside all the literature concerned with best practices in ontology building, development, discovery and the like. We rather consider the peculiarities of Semantic Web ontologies, in particular as regards their role, among other elements of the big picture of Semantic Web, in supporting automated information exchange. Then we discuss some weak points in that big picture that intrinsecally limit the possibilities of interaction of web agents.

Along the second line of evolution of the Web, we talk also about some peculiar tracts of Social Web and especially about the common practice of (free) tagging, both on data provided by users themselves and on data already available in the Web. We aim to show the real value that rough forms of catalogation of data such as folksonomies [Vander-Wal, 2004] can have in making the ideas and objectives of Semantic Web to scale to the dimension of the World Wide Web. In doing that we think that the words by Jim Hendler [Hendler, 2009] “a little semantics goes a long way” can be understood also in this direction – in spite of the fact that the phrase was to support at its beginning the effort in formalizing knowledge in ontologies for the Web.

At the same time, we try to narrow the distance and difference usually considered between ontologies and folksonomies. The final intent of the introductory part is to instil the doubt whether ontologies and folksonomies really come from different worlds and then suggest the possibility to deal with them on a paritary basis.

Because of the peculiar character of our project of thesis we demand the understanding of the reader for our choice to not account for too many details about the techniques actually adopted and the strictly technical aspects of the formalization process of information and knowledge in ontologies and the like. Indeed we prefer to pass over them and go to the core of the issue: the assumptions which are made by KR specialists and the expectations which are more or less explicitly said about the role of KR for the enhancement of the Web.

We hope nevertheless that the lack in details about state-of-the-art techniques will be considered as compensated by the original global account for the “operations” on ontologies that we propose. That is, we have noted a lack in the literature concerning the use of ontologies as instruments to support interaction as regards the theoretical account both of the role played and of the logical value of the processes of use of ontologies (mapping, merging, . . .). We then do it: we propose a reading of such processes that highlights the basic logical behaviour of operations on ontologies. Clearly, this moves also a step forward to the introduction of our proposals, as it allows to compare what is the “value of use” of ontologies for the Web with that of alternative structures.

Besides getting folksonomies closer to ontologies and *viceversa*, we conclude our preparatory part of the work with the (unsurprising) result that the bottleneck of Semantic Web is the foundamental reliance on a “linguistic paradigm”, which is shown in particular in the stack of languages needed to describe lower levels (the many *meta* that pile up in order to properly manipulate what is below) and which causes the need for translations. Indeed, in spite of a purely scientific context where one metatheory, once defined, can be generally accepted by the community and pretty quietly adopted, the Web is a highly dynamical environment where no ontology can claim to be THE ontology to which any other should conform. Here it comes the need for translation from an ontology to the other in order to have agents sharing knowledge, exchanging information over the Web. And here is also the patent impossibility to have working translation rules for any two ontologies. The alternative, we argue, is to rely less on language, even though it costs quite a lot in terms of loss of precision. But the more precise a concept is, the harder is to communicate it in a different language. Once again we would say “a little semantics goes a long way”.

Our contributions

Instead of an approach oriented to (highly expressive) languages, in order to richly represent Knowledge – although they are never expressive enough if one minds also to keep the represented knowledge tractable by a computer – we look for an approach that goes below the language, to the most basic elements of knowledge, and is just able to provide a plain logical structure to information. We find an interesting candidate for that in the geometrical approach inspired by Linear Logic [Girard et al., 1989, Girard, 1987]. The idea is to leave aside the particular terms, and their intended meaning, used to call and define concepts in Semantic Web ontologies and just look to objects, to the resources and data which are collected under each of such terms. It also means to leave aside the set theoretical setting of the whole ontology-related matter since this last relies on it. Ontologies indeed can be observed and used from both the intensional and extensional aspect of concepts/classes (and the same of relations and their extensions), even though are there some eccentricities with respect to standard set theory, like for instance the fact that collections of resources in knowledge bases are not in the general case sets as set theory would ask (consider for example their variability over time); or the fact that intensional considerations (like the defining axioms of concepts) could clash with, and prevail on, the extension principle.

Thus, in lieu of sets and set theory we find logico-mathematical reference objects in Girard's Coherence spaces [Girard et al., 1989]. A Coherence space is given by a collection of points and a binary, reflexive, symmetric relation, the *coherence*, on them. Any group of points all mutually coherent is called *clique*. We borrow such objects from Linear Logic's denotational semantics and use them to represent ontologies, with a subtle change in terminology, since the term coherence seems to us too delicate to be used in Semantic Web, and coherence is not actually what we would consider. Then we call them *Compatibility spaces* for the structuring relation is rechristened *compatibility*, since compatibility, rather coherence, is what we want to consider, observe and record among web resources.

Now, an ontology is represented by a Compatibility space and any concept in it is represented by a clique, that is all the objects that can be considered compatible under certain respects. It is really the same idea that underpins concepts in standard ontologies, where some quality or property (the ontological essence of the concept) is shared by all the members of the corresponding class. It is just made more clear the fact that all the elements (our points in the clique) are compatible, interchangeable for that respect, and less apparent – really it is hidden – the quality itself, the essence instantiated in those elements, since it is given no name at all. But that is the cost of reducing the prominence of language and making room for the observation and study of interaction at a more basic, less constructed and artificial level.

Once concept names and concept definitions have become invisible, we have no more difference between ontologies and folksonomies. Or, to be more precise – it sounds like paradoxical! – folksonomies are now richer structures than ontologies since they have a dimension that ontologies miss: the social aspect. Indeed, a folksonomy is the union of many personomies, each one being represented by its corresponding Compatibility spaces. Cliques here, in a personomy, represent concepts belonging to just one user registered to the social environment from which a folksonomy results. Such cliques reflect the user's own findings on compatibility among webpages, documents, photos and the like. The folksonomy is then a larger compatibility space whose support is made of all the points occurring in at least one personomy-corresponding Compatibility space and where new cliques may appear. These last will be made of n points pairwise compatible for $m \leq n$ different reasons, up to possibly one for each involved personomy. But for all these points a common essence, or quality, or whatelse can be considered, even though nobody in the community had found it before, for in any case there exists some question to which any of them is a correct answer.

The result will be then a change from general and well defined knowledge, but useful only for the people that already know that (since they all speak the same language) to only locally meaningful information, but the use of which turns out to be more informative, more instructive when communication successes. The crucial point seems to be precisely such locality, which is also the point for which we call into play Ludics [Girard, 2001].

A personomy, whichever be the way how it is represented, is significant just for the user u that creates it. But also for other users in the same community that observe which kind of resources u tags and which tags she uses. They could also (at least in the applications that offer such facility, but in principle we may assume that they can) adopt one or more tags as they are used by user u, and possibly define to which of their own tags they are to be associated, thus affecting also the resources that bear the involved tags. By the way, this provides us with a much more direct, explicit hint to consider a possible mapping from a personomy to another one than many of those on which rely standard ontology "operations". The challenge at this point is how to conceive the same processes that we have discussed about ontologies now that we have put on the same level ontologies and folksonomies by removing from ontologies most of what made those operations hold. On the one hand we have, as a result of our discussion on ontology operations, that really are into play only few mechanisms, which substantially reduce to logical inheritance and cut (the rule from sequent calculus). On the other hand, like operations on ontologies end up in special relationships among the involved ontologies, so once we are dealing with Compatibility spaces the same relationships can be observed over their points and cliques. What is missing for the moment, as well as it is missing for ontologies, is a general theory that indicates how and when two ontologies can be mapped or merged. That is, precisely on which terms, to which extent. And it is something that requires to know the intended meaning of the concepts with respect to the real world, the particular domain that the ontology is concerned with. This is the reason why the issue is faced by human experts – possibly aided by text- and datamining tools that carry out the most boring part of the work.

Nevertheless, with respect to the Compatibility spaces we can say something more. Indeed we are forced to not consider intended meanings (we are blind to them in the geometrical setting), and to look only at data actually available, grouped into cliques according to personal tagging. Ontologies are reduced, on this respect, to single personomies (the *shared conceptualization* of which a well known definition of ontologies speaks ([Gruber, 1995]) indeed expresses just one point of view). In order to look at resources whose types are not yet specified (we can just see groups of objects) we finally rely on Ludics. Its capability to discover whether two objects are equivalent based on the observation of their interaction is not requested here to build cliques within single personomies – this should be done just by directly registering users' tagging activity. Ludics rather is precious for the capability to trace and model the dialogue between two agents (in the most general sense) that interact in order to prove and accept some assertion, or to prove the contrary and reject it. Obviously such a capability fits also for the case of web agents. Then, based on Ludics, and on a couple of previous experiences that have succesfully adopted it to unveil particular dynamics in the dialogical phenomena (such as [Lecomte and Quatrini, 2009] in linguistics and [Fouqueré, 2011] in describing the client-server interaction at the basis of Wold Wide Web functioning) we propose a model of the interaction of web agents on resources represented in form of Compatibility spaces. The Ludics dialogue in this case is a paraphrase of a query-answer session in which the roles of questioner and respondent invert at any step of the communicative exchange and the dialogue can flow freely, unconstrained by any predefined protocol of communication – that is no pre-formed query schemas. Some particular cases then, such as that of users in a community that re-use other users' tags, let us also present the most interesting part of the whole story: how the contact can be established – that is how communication between two Compatibility spaces is activated (*via* the discovery of a cut somewhere) and how inheritance lets to propagate information (knowledge) from a source to the other one, and consequently from an agent to the other one.

Plan of the work

The thesis is structured around three main parts. The first part is dedicated to an exhaustive introduction to the two parallel lines of development of World Wide Web, that is Semantic Web and Social Web, or Web2.0. In particular, is conducted there a (critical) analysis of the overall project followed within the Semantic Web initiative (promoted by the World Wide Web Consortium) precisely to achieve Semantic Web; whereas a comparison between the peculiarities, which reveal complementarities, of the approaches adopted by the two evolving lines leads to the conclusion that they can be the two faces of a same coin and should be exploited together, in a strict collaboration on the part of the communities involved but also on the basis of a suitable logical framework.

The second part deepens the analysis on ontologies, as the main form of knowledge representation for Semantic Web, and attempts an unprecedented comprehensive description on logical basis of the typical "operations" on ontologies (the processes of use that allow to compare and/or compose different ontologies, so as to support interoperation between information systems), which are normally described just in terms of the algorithms adopted for case-specific applications. Then, in this part are also introduced our contributions, that is first of all the representation as Compatibility spaces of ontologies, but also of folksonomies and, in principle, of any form of information or knowledge representation – specified on some data – suitable for the Web. Subsequently, it introduces also our first attempt to describe the interaction of web agents through the dialogue-like structures of Ludics, which are grounded into Coherence spaces (of which our Compatibility spaces are slightly more than a terminological variant).

The third part finally gathers most foundamental stimuli, observations and remarks from the previous parts of the work and articulates them into a more philosophically aware discussion. The peculiar relationship between ontology and language, as well as the compresence of multiple ontologies or the social (community related) aspects of ontology generation, among other delicate issues, are discussed there, just after a brief history of ontology that only touches some significant steps that we isolate to point out an evolution of the approach to and use of ontology in philosophy, science and the society over centuries, from Aristotle to nowadays.

Along with interesting results concerning the observation and description of interactive processes over the Web, the research line opened with this work leaves also a number of aspects to be furtherly investigated. First of all the role of Ludics needs a more detailed framing. In fact, at present the setting of Ludics-like modelling of Web agents' interaction is somewhat provisional and could require some changes to better describe it. Also it should be considered in a wider range of cases and could be compared to other approaches to the description of agents' interaction though not specially committed to Semantic Web (like in [Cardone, 2011]). As regards Compatibility spaces, should be considered what may be the value and meaning, with respect to ontologies, of the other operations that can be performed on Coherence spaces, like e.g. the use of linear logic exponential operators. Then, strategies to detect "interesting" cliques within a Compatibility space – e.g. cliques representing new emerging properties not accounted in the corresponding ontology – would also deserve some attention. Last but not least, even the philosophical considerations, that at present are proposed as punctual stimulating observations in response to challenging issues that have emerged during our work on KR ontologies, could be taken into deeper analysis and higher speculation, and organised in a broader, global philosophical study of the matter.

Please note that the references of all the parts of this work (this Introduction and Parts I, II and III) are given separately at the end of each part. The overall bibliography is given at the end of the book.

Bibliography

F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logics Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

Tim Berners-Lee. Issues for the future of the Web. Opening keynote at the First International World Wide Web Conference. `http://www.w3.org/2001/sw/Europe/graphics/tmbltext.html`, May 1994. Geneva.

Felice Cardone. The geometry and algebra of commitment. In *Ludics, Dialogue and Interaction*, volume 6505 of *FoLLI LNAI*, pages 59–79. Springer-Verlag, 2011.

Christophe Fouqueré. Ludics and Web: Another Reading of Standard Operations. In *Ludics, Dialogue and Interaction*, volume 6505 of *FoLLI LNAI*. Springer-Verlag, 2011.

Jean-Yves Girard. Linear Logic. *Theoretical Computer Science*, 50:1–102, 1987.

Jean-Yves Girard. Locus solum: from the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, pages 301–506, 2001.

Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types*. Cambridge University Press, 1989.

Tom Gruber. Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human and Computer Studies*, 43(5-6):907–928, November 1995.

Jim Hendler. "a little semantics goes a long way". `http://www.cs.rpi.edu/~hendler/LittleSemanticsWeb.html`, 2009.

Alain Lecomte and Myriam Quatrini. Ludics and Its Applications to Natural Language Semantics. In *Logic, Language, Information and Computation. 16th International Workshop, WoLLIC 2009. Proceedings*, pages 242–255. Springer Berlin Heidelberg, 2009.

Barry Smith. Formal ontology, common sense and cognitive science. *International Journal of Human-Computer Studies*, 43:641–667, 1995.

Barry Smith. Against Fantology. In Johann C. Marek and Maria E. Reicher, editors, *Experience and Analysis*, pages 153–170. HPT&ÖBV, 2005.

Steffen Staab and Rudi Studer, editors. *Handbook on Ontologies*. Springer Berlin Heidelberg, 2nd edition, 2009.

Thomas Vander-Wal. Folksonomy. `http://vanderwal.net/folksonomy.html`, 2004.

Part I.

Structuring Web Knowledge: ontologies and folksonomies

What is happening on the Web today

In this part we will illustrate what is already there in the Web, focusing on that which is aimed to the improvement of the World Wide Web itself. We distinguish two main streams in which the efforts of different communities have been gathered for more than ten years with the common target to improve the Web: the first current, that follows a mainly technical approach, is led by the World Wide Web Consortium (W3C) and is known with the name of *Semantic Web Initiative* (SWI); the other one, which flows spontaneously thanks to a large number of people that simply care about a better experience of the Web for their own sake, is widely known as the *Web2.0*.

Semantic Web (SW) is intended to provide more structure to data available in the Web so as to allow computers to "understand" the information in the Web and provide it to human users somehow already "digested" – whilst nowadays e.g. after querying a search engine (like Google) you have to read the pages that have been retrieved as responding to your query in order to actually have the answer (the precise information) that you are looking for. Since the task of structuring data for use by machines is not a real novelty, the Semantic Web Initiative has co-opted the research area of Knowledge Representation (KR) to recover suitable models and formalisms and then operated over them a double work of adaptation, to be compliant with a networked environment, and of standardization to facilitate large-scale diffusion. Even though the models and formalisms chosen from KR experience are the result of a long time of research and development, we have some questions about the possibility to really build the Semantic Web (only) based on them.

Conversely, Web2.0 does not follow any specific project for the improvement of the Web, it simply produces it *via* the contribution of millions of users of the Web that provide first of all data, and then also some level of "semantic enrichment" of data and resources (provided by either themselves or anyone else), especially using the easy technique of tagging. Finally, they also contribute in structuring the Web linking resources over the net, thus creating communities both of users and resources, aggregated according some peculiar (or maybe idiosyncratic) criterion. Even though no assumption can be made about the reliability of structures emerging from such methods and techniques, they have proved to be effectively useful (cf. for instance [Brooks and Montanez, 2005]) to enhance accuracy in Web searches.

In what follows of this first part of the work, while describing the project of Semantic Web and illustrating key-aspects of the relevant ongoing research, we also present our position about them proposing two kinds of arguments: technical and conceptual objections against the way how data structures should give meaning (a semantic) to data, and remarks about the role that the users of the Web should play to achieve the new Web. In fact, on the one hand we point out, at a theoretic level, some weaknesses of models and formalisms chosen to represent (or construct) data structures in the Web, which finally show that they are not flexible enough to cope with the extremely heterogeneous, quickly-changing and, in a word, open environment of the Web. On the other hand we complain the fact that the Semantic Web Initiative takes into almost no consideration the other path of evolution of the Web, even though, we argue, some of the best practices emerged in the Web2.0 could be profitably exploited to effectively manage large amounts of resources (web-pages, media files and everything that can be located on the Web). By the way we note that scalability is precisely an issue that raises problems to Semantic Web, so that the best that one could imagine is the Semantic Web experts and the Web 2.0 enthusiasts to join their efforts and mutually get advantage of the others' experience.

While introducing Semantic Web and Web2.0 (or Social Web), we will also focus on the most representative models for structuring information that they have developed, that is respectively ontologies and folksonomies. Ontologies, to be honest, are not an original idea of Semantic Web – rather they are well rooted in the area of Knowledge Representation – but get a new look when used in the Web precisely for the open nature of such an environment. Semantic Web and ontologies then propose an institutional, strictly top-down approach to the final objective of improving the Web. On the contrary, Web2.0 and folksonomies propose an approach completely different, radically bottom-up, with folksonomies which are a pretty new, original product of the Web2.0 – and whose real cognitive value perhaps has not yet been fully understood. Our comparison between Semantic Web and Web2.0 therefore will be mainly aimed to identify the differences related to, and dependent on, these alternative cognitive frameworks, especially by considering their value as structures with a logical meaning, that is, for which semantics can be given, and therefore suitable to be directly used by computer programs. But we will also look for the similarities which will provide a common ground to settle another kind of cognitive structure, another way of logically representing information and knowledge in the Web.

As it will become clearer after the following three chapters, our intent is to bring Semantic Web closer to the evolving Social Web using directly the knowledge put into Web2.0 environments (e.g. tagging spaces) without the steps of formal adaptation to make it compliant with the W3C standards and the canonical way how Semantic Web is expected to be achieved – steps that require an effort in representation and validation of that knowledge which simply could be avoided. Briefly, we think that the technical enhancement (Semantic Web Initiative) and the social movement (Web2.0) are two faces of the same coin. Therefore the community that is developing the technological layer of the Semantic Web should strictly cooperate with, and take advantage of, the Web2.0 communities that spontaneously provide the Web with collections of resources that are classified (though roughly) according to some collectively developed knowledge framework.

1. The World Wide Web technical enhancement

In 2001 Tim Berners-Lee, the inventor of the World Wide Web together with Robert Cailliau, presented his vision of the Semantic Web in a divulgative article on Scientific American [Berners-Lee et al., 2001]. It depicts an almost science-fictional scenario where one can ask her PC to arrange some sessions of physical therapy, in a center covered by her medical insurance and not too far from her house, accordingly with the other commitments scheduled during the week. And the computer readily proposes some possible solutions asking the woman to indicate which solution she prefers in order to set the arrangement. Beside the fictional aspects of the story, that article exposed to a quite large audience the idea of Semantic Web: apparently the same Web that everybody knows but where also computers can understand the information that it contains, thus enabling them to perform some quite complex tasks like arrange for a physical therapy or reserve trains, flights and hotel for a scientific conference in a foreign country, without the human users have to spend their time in face of the screen looking for the cheapest offers on many different websites.

In a previous publication, the book *Weaving the Web* in 1999, Berners-Lee had already spoken about the Web, from its origin up to the future development, even with some criticisms about the way it has been produced, especially as regards the imparity between the ability to read, or use, the Web and to write, or build, it. Berners-Lee points out in [Berners-Lee and Fischetti, 1999] that guilty for this lack, really a missed opportunity, are those who developed and commercially diffused the first browsing softwares, because they separated what in his mind – and also in his prototype browser – had to be a single tool to both follow links on the global hypertext and build new links either between already existing pages or towards new ones just written, so as to add them to the web of the hypertext.

Most of all in [Berners-Lee and Fischetti, 1999] Berners-Lee clearly affirmed that the World Wide Web – at that time – was still far from its mature form, which should have been something closer to what he later describes in the article on Scientific American. That is, since quite early, maybe even short after its invention if one fully trust in Berners-Lee's words, the Web was intended to be much more than an hypertext that people could read and compose: its great potential is that of allowing computers to use the information put into pages and not only to display it on a monitor acting as mere access terminals. In fact, at the time of *Weaving the Web*, the W3C was already working on RDF (see section 1.2), one of the main components of Semantic Web, and had already developed XML (in 1998), the first language for structuring Web documents, thus showing a clear interest to give logical structure to the information beside formatting instruction for on-screen rendering.

Now, over ten years after that book, we cannot see great difference. Surely W3C has got many important results in defining standards for other components useful to achieve the Semantic Web (mainly the stack of languages to define data structures, together with a query language); there are also many more researchers, both academics and from private companies, that work more or less directly for the big project of the SW; and finally one can easily find some Semantic Web "oasis" sparse in the Web, that is some quite restricted groups of datasources that can actually share data thanks to the composition of (pretty *ad hoc*) interfacing and querying services, precisely in the style of Semantic Web. But we are still very far from having something similar to the (still science-fictional) article from 2001. Most of all, we lack the spread diffusion that can be achieved only by means of large contribution from common people, i.e. everyday Web users – like it happened (and it's still going on at present) for the first period of the (simple) Web when a constantly and quickly increasing number of people with no special formation discovered how easy it is to add a piece to the global hypertext.

Indeed we have at present a huge number of persons that publish pages, images, videos, any kind of resources on the Web, but for the most this happens under the flag of Web2.0, not in the specific way Semantic Web demands. This is not a matter of party, it's a practical one: Semantic Web achitecture requires to adhere to certain standards and needs to reduce the different ways to describe data to the lowest possible number, but this implies also i) that people know which are the standard languages and how to use them, thus requiring some level of technical competency that is quite higher than needed for simple HTML formatting[1]; and ii) that people accept some set of particular data structures to describe their own resources. This latter is not a minor point since together with such data structures one finds also quite fundamental ontological assumptions about the world. Therefore, to have people properly using them, it needs that they know both the data structures and the assumptions on the world that these endorse. Otherwise, Semantic Web will stay a technical matter only for experts. In what follows of this section we will show what in the Semantic Web project causes these hindrances on the way to SW.

1.1. The Semantic Web Initiative

To introduce the Semantic Web project we can read first of all what the W3C says about it:

> The Semantic Web is the extension of the World Wide Web that enables people to share content beyond the boundaries of applications and websites. It has been described in rather different ways: as a *utopic vision*, as a *web of data*, or merely as a *natural paradigm shift* in our daily use of the Web. Most of all, the Semantic Web has inspired and engaged many people to create innovative semantic technologies and applications[2].

[1] Even though this is not so a big problem since convenient tools could be provided to help in the task, we have that no such a tool is available, due to the fact that the deeper questions like those involved with the following point should be resolved before.

[2] From the Semantic Web Initative wiki at `http://semanticweb.org/wiki/Main_Page`

What seems to be the most interesting aspect here, beside the awareness of a utopical appearence, is the focus on data (a web of data) and on the contribution of people who have developed "innovative semantic technologies and applications". The vital need of Semantic Web for contribution by people is officially stated as regards the use of data thanks to semantic technologies, while little attention is paid to data production or provision – that indeed is paid a larger attention by Web2.0.

We can also try to say it another way, our own: within the Semantic Web Initiative it is assumed that expressive formal description of data sources will lead to their interconnection throughout the World Wide Web *via* logical interdefinition of concepts appearing in different descriptions. Now, to make it a little clearer, this means that the basic idea of Semantic Web is to enrich with special descriptions possibly all the resources available in the Web (that is, provide with *metadata* all the data). In order to be understandable to a machine, however, such descriptions must match some other description that defines the meaning of the metadata (if one thinks of meta-metadata he's not far from what's in the game, even though officially that is called the *schema*, or *vocabulary* or *ontology* depending on the richness, accuracy and logical formalisation of the description); but one cannot expect that a computer knows (has in its memory) all such schemas once and for all, so it is possible to link descriptions each other in order to allow for (mutual) inter-definition of the element types (above referred to also as concepts) appearing in different descriptions. If the picture is clear, one can now understand how every datasource exposed in the Web could provide easy access to its data to *autonomous web agents* surfing the Semantic Web. Web agents will be main (though hidden) players of future Semantic Web: they are software programs launched from a user's machine whose mission is to search and retrieve pieces of information or directly accomplish some task, like in the example from Berners-Lee's article. After the flop of classical Artificial Intelligence[3] (AI) has shown that teaching everything to a machine is not a viable way, AI researchers have the opportunity within Semantic Web to deal with multiagent systems where each agent knows very little about the world and "learns"[4] what it needs by looking for ontologies – where it can discover the meaning of data through that of relevant metadata – thanks to the ability to follow the inter-definition links among ontologies.

Thus Semantic Web expects not only to provide rich descriptions capable to teach the meaning of data to machines, but also that such descriptions are given a formal, logical specification (a formal semantics of data) and that any single structure can be linked to any other, exactly as the simpler hypertextual pages are linked each other, so that one could expect some day we will have a global structure describing everything in the World Wide Web. Indeed, even though actual W3C expectations are quite humbler, that is the idea. In such a way, while crawling the Web to accomplish some specific task, a web-agent faced to some data whose meaning it does not know, it can move from the file that formally describes those data to any other description file

[3]We accept the distinction between a classical AI, which aimed to reproduce the high-level reasoning abilities of human beings, and a range of different approach to low-level intelligent behaviours, which are often compared to animal abilities, such as cybernetics for instance.

[4]Machine learning may or not be an element in Web-agents, however that does not make the point here.

which is linked to the first one until it meets some known concepts and eventually gets the meaning. Or in other words, the agent can search for something that uses terms that it already knows to explain the formal semantic of a new term (or concept) that it has just met.

According to these core ideas, the research area of Knowledge Representation (KR) has been co-opted in the Semantic Web Initiative. Formal, logical definition of knowledge is indeed a task on which KR works since long ago, and to present knowledge about the information stored in datasources in a way that is not only machine-readable, but also machine-understandable is exactly what in the big picture W3C needs to make Semantic Web run.

Like in databases the meaning of data is given by the logical schema of the database – which acts as the context of data and may be realized in the form of a tabular structure with column captions (the names of the attributes that compose a relation) – metadata in the web provide the "tabular" structure that binds data giving them context and thus making it possible to recover information from them. But another level is required (remember of meta-metadata) to have inter-operability over data, i.e. the possibility for any Web-agent to find information wherever in the Web. This further level is precisely where KR comes into play. Indeed, whereas the context of a logical schema of data lets the data represent information for a computer within a specific database system[5], to make such information understandable to any other machine (represented by some web-agent) even outside this special situation requires to provide a higher level context to the information (thus passing through the second step in a knowledge chain that may look like this: data → information → knowledge). This means not only to name the elements that correspond to portions of data (as it is done with captions in the columns of a table), but also to define what "entities" they are, to give them a formal, possibly logical definition. Knowledge Representation then operates here to provide the suitable descriptions. Its involvement however offers also the ability to execute automatic reasoning – the current way to propose last developments from classical AI research – so that rich (and sound) descriptions enable inference drawing. W3C has led the merge of the most promising KR models (namely ontologies) with certain of its standard markup languages for the Semantic Web (some of OWL's dialects) so that, by adopting such languages, one can produce a typical Knowledge Base (KB) – in the full sense as in KR tradition (see chapter 2 for further details) – where an inferential engine (automatic reasoner) can be easily plugged in. Finally, the interconnection of datasources in the Web turns out to depend on their specification and inter-definition in a formal way suitable for automatic reasoning.

Now, about over ten years after the adoption of KR to support the building of the Semantic Web, it is apparent that it is still a long and hard way to walk. Indeed, if considered as a research program it looks like being too ambitious and expects great efforts by the side of data "owners" and content providers, who have to realize the metadata layer(s); and if considered as an industrial/commercial initiative (as it is at least partially) it follows a completely top-down and anti-dynamic approach that is

[5]We mean in general the common closed situation (as opposed to the open environment of the Web) of a system where the meaning of data is defined – and all the software that could ever use the data is also *ad hoc* programmed – once and for all (at least in principle) by the system developers.

opposite to the "nature" of the Web. Top-down since ontology definition is invariably a descriptive operation that freezes the knowledge about something according to its understanding on the part of a restricted group of experts. Indeed, even though it has to be a shared conceptualization [Gruber, 1995] it is nonetheless the result of a negotiation among specialists, i.e. an imposition of a particular vision, possibly a very reasonable one, that is to be taken as it is. And anti-dynamic since once an ontology is defined there is no way for a user to change it. On the contrary, the authors of the ontology could even change it, but at the price of noting it as a different, newer "release", that is another ontology.

The point is that in order to have knowledge representations suitable for use with current technologies and according to the theoretical background of Knowledge Engineering (KE, that is KR plus reasoning), we need to already have data and then describe them – or at least to know *a priori* which types of data will appear in the datasource we are going to describe. This is not necessarily the general case for the Web, where the publication of data and resources gets more and more driven by users. The point gets even more complicated in the frame of the *web of ontologies*. We have mentioned the web of ontologies as an almost natural linking of ontologies when they appear in the Web, but reality is quite different and that vision is pretty utopical. For instance, we already have lots of ontologies and most of them share absolutely nothing[6] and the only way to create a web of logical and meaningful inter-definitions is *ad hoc* mapping between ontologies. This is indeed the actual way how ontologies are made to communicate. We remark that such a way requires that are given in advance both formal descriptions of the datasources and the rules for the mapping between datasources to have them actually exchanging data and resources. In particular, the mapping issue prevents even to compare any two datasources unless the mapping instructions have been previously provided for the ontologies that describe them.

In order to fully point out strong and weak points of the project of Semantic Web, as it is led by the W3C, and to justify our criticism and objections we will now alight on the techniques which have been deployed for its realization. We will progressively focus on the W3C standard languages; the KR model of formal ontologies; the reasoning services on ontologies; and finally the "composition" of ontologies in the open space of the Web and their possible use on the part of Web-agents.

1.2. Standard languages

The typical W3C multi-layered approach to Semantic Web is perfectly shown in the following figure 1.1. The name of "Semantic Web layer-cake" comes directly, as the figure too, from W3C presentation documents about the Semantic Web Initiative. The version we propose here dates at least 2007 and slightly revises a previous version of the picture that paid no attention to some aspects like e.g. encryption. However, that point is not so interesting to us, whereas our attention will be aimed to the layers

[6]This is one of the reasons why "upper" (foundational) ontologies have been conceived, but apparently they do not close the issue.

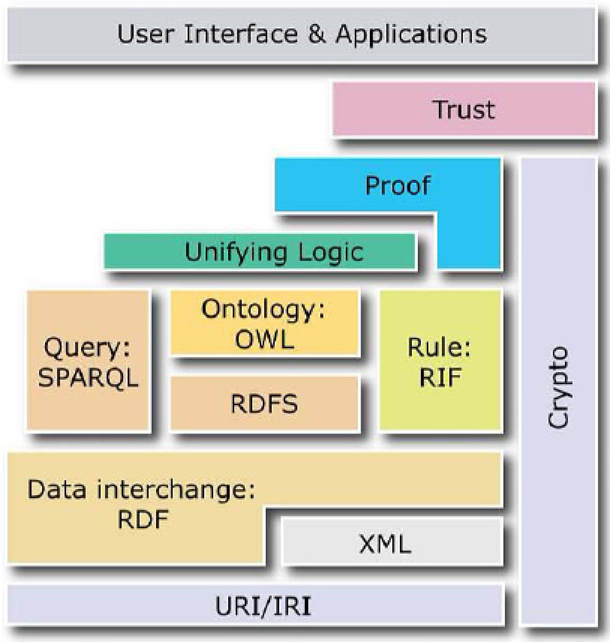


Figure 1.1.: The Semantic Web layer-cake

called Unifying logic and Proof, but before we reach them it is worthy to face step by step the lower layers.

Semantic Web is rooted in nowadays World Wide Web by means of the use of the same specifications to identify objects (resources) in the Web, i.e. URIs (Uniform Resource Identifier) – along with the extension IRI (Internationalized Resource Identifier) that enriches the alphabet for legal URIs from the original subset of the US-ASCII to the Universal Character Set (Unicode/ISO 10646) that allows to write URIs in any language. Common URLs (Universal Resource Locator) used to reach resources in the World Wide Web (documents, videos, photos, everything we can point our browser to by writing its address in the address bar) are just a subclass of URIs, and quite often also SW resources are named with "working" URLs, i.e. URIs that also locate some resource (basically a text file) in the hypertext. Apart from this fundamental common element, Semantic Web and World Wide Web get progressively more distant while getting higher on the layer-cake due to a radical difference about what they are intended to offer: World Wide Web is a huge hypertext of which we can reach and see (displayed on screen) any single page; on the contrary, Semantic Web aims to stay hidden – so no visualization on screen – and describe to machines (software programs) the meaning of the data that are published at some location in the Web. As a direct consequence we have that (simple) World Wide Web needs only a standard language to determine how things should be displayed on screen and to follow links from page to page, a task that is accomplished by HTML (Hyper-Text Markup Language)[7] combined with the original HTTP (Hyper-Text Transfer Protocol) protocol to move from page to page. On the contrary, Semantic Web (in W3C view) has to climb a stack of different layers and specific languages to provide a super-structure capable to tell a machine what the content of web-pages means.

1.2.1. Structuring data with xml

The first step to provide logical structure to Web documents is the ability to mark portions of information within a document, of which one can then explain the meaning. That of marking portions of information is the main functionality of XML (eXtensible Markup Language). As its name already says, it is a mark up language[8] devoted not to formatting for on screen presentation but to delimitation and identification of text (data) blocks that constitute information units, and it is not intended to be displayed to human users since it "speaks" to machines. HTML is a complete language, whose tags[9] are all (pre)defined once and for all, that is it has a fixed syntax and a fully fixed semantic. This offers the World Wide Web the ability to be accessed, viewed and used exactly in the same way (at least in principle) independently from the particular browser, system and machine one is using to browse the Web. On the contrary, XML is rather a grammar (or a meta-language according to W3C naming fashion) that allows to develop other "private" languages so that one can use XML to build a special language for her/his own purpose. The definition of such a private language passes through the listing of all the terms that your particular XML language (called an XML application) will use. It is actually something like giving an additional vocabulary to a pre-existing grammar – that is the reason why it is called *extensible* language – that allows to identify within compliant documents any part of information that the language developper considers worthy to be isolated.

Let's spend some more words about how all this happens, since the mechanism will be the same also for upper level languages in Semantic Web stack. Since XML is intended to mark the portions of information in the documents that are produced in some particular context, for which a particular XML application is developed, the vocabulary must list any type of "element of information" that one expects to deal with within her/his application – each entry of the vocabulary being an XML element that could be found in the XML documents compliant with that XML application.

[7]Other languages to create so-called dynamic Web-sites, such as PHP and ASP, are a matter of ease of building and maintain of web-sites and do not affect the basic working of the World Wide Web.

[8]Like in most Semantic Web layers, we have no programming strictly speaking, but only markup and formal descriptions or definitions. Nonetheless, programming is a counterpart that is implicitly assumed in order to have services (web-agents) actually crawling and using the Semantic Web. In our opinion, and this is what we aim to with our research, Semantic Web programming should be extremely light-weight, that is assume only minimal knowledge behind an agent – compensated by a great ability to discover new knowledge.

[9]In HTML a tag is a marker that dictates the browser how to render on the screen the text that it comprises. Normally it appears as a couple of delimitors (opening and closing tags) enclosed in angular brackets, which together mark the start and end point whithin where the text is to be rendered in the special way indicated by the tags. Any part of a web-page is enclosed in one or more, possibly many, HTML tags.

For each XML application we have then a single document where the vocabulary is introduced – that is called the schema[10] – and a class of other documents which are all the documents that are written according to that particular language. Once a "personal element" is created within a XML-schema, it can be used as a couple of tags (the opening and the closing one) as one does with HTML tags. XML-schema offers also some structuring facility that allows to give structure to the vocabulary (as if suggesting some deeper relation between elements of information) by defining nested elements, that is elements that are part of another broader element – that which implies the building of a hierarchical tree structure – and attributes, that is portions of information strictly related to some particular element. Together with the listing of valid XML elements for a specific application, the XML-schema provides also some complementary information about each element and attribute, such as the possibility to have no value (that is to say an optional portion of information) or the restriction for possible values to a special type of data (datatype) which may be for instance alphanumeric strings, date-and-time in a particular format (e.g. yy-mm-dddd@hh-mm-ss), a boolean (yes or no) or even a personalised datatype (a set of values defined in the same schema).

If one minds it better this is not really a vocabulary – in fact W3C calls it just XML-schema – since it provides no description nor definition of the terms that it introduces. It just establishes which elements do exist for an application (and which attributes apply to each of them). The application is an instance of XML language, but it also implies a, maybe very rough, abstraction of a particular context of work on information, of the use of information that is done by a particular community of user for which the application is developed, what we can call a particular domain of interest. Precisely to provide richer descriptions of the domains, in order to compose good definitions for the entries of vocabularies which could even "teach" something to computers about their meaning, the other, upper level, languages have been developed at W3C. Nevertheless XML – whose first version dates back to 1998 – has already been largely exploited for important operations that generally can be seen as translation operations. When the semantic of the departure language is known and that of the destination language too, XML allows for a clean translation of any document written in a (meta-)language into the other. It needs only that skillful experts define all the translation rules. This is what happens quite often for large databases integration, e.g. when a big company acquires an other one. In every such case, there is a little number of languages[11] involved (typically just two) and it is worthy (even advantageous) to manually produce the

[10] Besides XML-schema there is also another, previous way to define an XML application, called DTD (Document Type Declaration). Such a solution however presents some important differences with respect to XML-schema, ranging from datatypes handling, i.e. the precise definition of the set of values that an element or attribute may accept, that is very poorer with DTDs, to the limitation of allowing a document to depend on only one DTD, whereas multiple XML-schemas can be recalled within a single XML document. By the way, general Semantic Web orientation is precisely that of multiple references to many different vocabularies; it is not by chance that XML-schema comes later and aims to replace DTDs. Finally we may note also that DTDs are not written in XML, whereas XML-schemas are.

[11] As a language here is to be intended a particular information portioning specification, like e.g. the logical schema of a database, which corresponds to the structuring capability of XML.

translation rules – while the alternative would be re-enter a huge amount of data. By the way we note that the same does not hold in the open space of the Web, where the number of languages is not *a priori* known and possible translations needed become soon an unmanageable issue.

We have to spend some other words about XML to introduce the *namespace* mechanism. Conceptually a name space is the place where the elements of an XML application are "created" and exposed on the World Wide Web so as to be publicly available. A namespace then is the identifier of that place. It coincides with the URI of the XML-schema and normally it is a URL. Thanks to namespaces we have a simple but powerful mean to re-use tags (i.e. elements, or generally speaking terms) defined elsewhere and wherever in the World Wide Web. While parsing an XML document, namespaces point the parser to the XML-schema where the relevant XML elements are introduced (as element types indeed), thus allowing to validate the document (that is to check whether it respects its own grammar) which is a preliminary step in order to actually use the information in the document. Finally, the grammar to be respected is given in the tree-like structure of the relevant XML-schema, as a list of all the elements that pertain to that XML application, together with their possible attributes and specific datatypes (i.e. the special kinds of data strings that are allowed to fill in the "fields" of such attributes in an XML file intended to be compliant with that application). In the XML-schema, indeed, all the elements relevant to an application are organized within a hierarchy (the tree structure) with a single root element and arbitrary node elements, arbitrarily nested, each one possibly characterized by some attributes. The tree-like structure is another key element of information structuring (cf. section 3.1) in the Web and is also the main ingredient for enabling the inheritance mechanism that, in a pretty similar way as in object-oriented programming, allows to re-use already introduced elements (element types) in order to build other, more specialized, elements (element types).

Once the basic principles of XML are clear, we already have accounted for a great part of Semantic Web, since also the other, upper level, languages all share the same basic idea: after a vocabulary is specified, any document in the Web may recall it and use the terms (elements when thinking of XML) that are defined there, provided that it respects the intended meaning and obeys the formal restrictions set out in the vocabulary. As it will be clearer later on, the first condition is quite harder than the latter to be verified.

1.2.2. Describing information with rdf

With RDF (Resource Description Framework) W3C has walked a major step towards Semantic Web. It is mainly thanks to RDF indeed that one can speak with some reason about semantics in the Web – although even the general idea of Semantic Web is somewhat risky precisely for the claims about semantics of data.

RDF has been released as W3C recommendation in 1999, even though that specification has been superseded by a new one in 2004. It offers a model for data interchange over the Web: more than a (meta-)language like XML, RDF still provides the ability to establish a set of types of elements and to recall them into many other documents that refer to the original specification *via* the namespace technique, but in addition it gives the (types of) elements a more defined logical structure. Moreover, it allows to formally establish relations between the elements, and also to recall such relations in other documents in the same way it happens for the element types. In fact, RDF introduces some very important conceptual novelties. First of all, the element types are considered as general types, each one associated with the corresponding class of objects of that type. As a consequence, in RDF one speaks no longer of elements, but of classes (to say types), and resources get typed by the name of the class they belong to. This means that one is no more committed on portions of information within a (mainly textual) document, but may use RDF types to describe everything s/he might want to. The possibility to have nested elements that compose a hierchical tree structure, then, is replaced by the two more effective capabilities of ordering the types as super-classes and sub-classes (what produces a more meaningful hierarchy that is to be read according to the IS-A relation)[12] and of introducing explicit relations among the types to introduce whatever other relevant relation between types (other than the IS-A). In the same way a RDF type gets its own name and is located within a hierarchy of other types when it is "invented" to describe some kind of information object, also RDF relations (actually they are called RDF properties) are logically structured objects: they are given a name – what allows to recall a relation in external documents – and domain and range types are specified (like for mathematical functions), so as to define between which types it holds. While in XML the idea is to define a data structure for web documents to be strictly respected in order to have them properly recognized and used within a special application, with RDF one has a repertory of types quite richly characterized and relations among types that can be freely used in whatever web document in order to describe their information content possibly to any machine thanks to formal specification of the elements. Really, it is not so immediate to have an RDF annotated document be "fully understood" outside some specially conceived web service but the (basically correct) idea is that more logical structure may ease the task of understanding the information content of web documents – and with RDF we can speak more generally of web resources.

The homologous of a XML-schema for RDF is a RDF-schema, but it is licit (according to W3C) to call it a vocabulary. To define such "user generated" vocabularies W3C has also provided a special standard vocabulary, still in RDF, named RDF-Schema (RDFS, mind the capital S to distinguish the vocabulary for defining and publishing other vocabularies from all the other vocabularies) where fundamental terms are defined, such as e.g. *resource*, *class* and *property*. This way, by recalling the standard RDF-Schema, one can formally define her/his own vocabulary – that is a description of the objects, and relations among them, that characterize the specific domain for which s/he wants to use the RDF application – by introducing some new terms and marking them as classes or properties, plus some other details, like e.g. range and domain restrictions over special classes.

For instance, in RDF-Schema (the mother of all RDF vocabularies) we find terms like `rdfs:Resource`, where `rdfs:` is the short prefixed form for the namespace of that

[12]IS-A relation, so called for the shortening of "X is a kind of Y", is the basic relation structuring hierarchies of objects (in the broadest sense) in most computer science applications, but also in common knowledge organizations, cognitive arrangements and philosophical speculation.

vocabulary and `Resource` is the proper name of the class that contains every kind of objects about which RDF can speak about; `rdfs:Class` is the class of all the (RDF) classes, and since it is something about which RDF speaks, it is also a resource – we will come back on this later on. `rdf:type`[13] is rather a (RDF) property (i.e. it is a resource that belongs to the class `rdf:Property` of all the properties), and it is precisely the property that expresses the belonging of a resource to a class, so that its domain and range are `rdfs:Resource` and `rdfs:Class` respectively. We remark that a name like `rdfs:Resource` is a URI, composed of the URI of the name space (the short prefix standing for the whole URI) plus a specific identifier, unique to each term in a same vocabulary. This makes it possible, for whatever term into an RDF document, to go back to the name space where it is defined.

This is the way how RDF makes easier the task, for a machine, of understanding the "meaning" of a document: once the computer (more realistically: an application) is aware of the standard semantic of RDF and RDFS, when it takes as input an RDF annotated resource[14] it can understand that it is a resource in the `rdf` sense and that it belongs to a particular class (in the `rdf` sense) of resources, and possibly that it is linked *via* some special properties (again, in the `rdf` sense) to some other resources, according to the specifications given in the vocabularies which the namespaces that prefix the RDF terms involved with the annotations point to. Then, obviously, we may (must!) discuss what issues it resolves to know that information.

The data model that RDF offers to represent information in the Web is that of a graph structure, built up of many triples, that is two nodes connected by an arc. Such triples derive from RDF statements. Indeed, both terms definitions in a RDF vocabulary and RDF annotations about resources have the form of simple statements, made of subject, predicate and object, where the subject is some resource[15], the predicate is some `rdf:Property` and the object is either a resource or a data value (i.e. a string, an integer, . . .) generally called a *literal.* Since the significant part is that of the graph, the syntax according to which a vocabulary or the annotations are serialized in a file is not very important, it is enough to have the parsing application be able to correctly recover the corresponding RDF graph. Thus we have four alternative syntaxes to write RDF: the most verbose is that in XML style; quite more compact and simpler to write down are N-Triple, N3 (Notation 3), and Turtle, the last two being credited to be kindly human-readable. All these have been developed or at least endorsed by W3C. Besides these "competitor" syntaxes, there is the normative one – which is not to be actually used in the Web – called Abstract Syntax. It is normative in the sense that the semantic of RDF is defined for it, and any other syntax must be able to reconstruct the same graph that would be derived from a document produced according to the

[13] The namespace here is that of the base specification of the RDF language.

[14] While for XML we had to speak strictly of XML documents, compliant with the specifications of an XML application, with RDF we speak more generally of annotations about resources, in a way that is precisely that of metadata: some additional information regarding lower level data, provided along with the data itself and without altering its original format.

[15] A subject resource may be both an instance of the class rdfs:Resource, the simpler and more common case, or a blank node, that is a resource that appears within a RDF graph just to act as a bridge to link other resources and is not *in se* interesting, so that it is not given any URI to be referenced from outside that graph.

Abstract Syntax.

Human readability is not a primary interest about RDF since it should be managed only by applications to enable content-aware web services. Attention paid to the human readability can be explained as a way to facilitate RDF spread, especially in its first period, as this could mean ease of use in writing both RDF documents and applications that use them. But it may also reveal some interest to preserve the ability for humans to read such documents and get a real understanding of what description a RDF vocabulary provides about some domain. Even though this point may look irrelevant, it will show its relevance at the upper level in the SW stack, where vocabularies become ontologies.

Before we pass to that level we have to remark some critical aspects of RDF semantic. As it is stated by the W3C in the apposite Recommendation [Hayes, 2004], besides any other possible way to define a semantic interpretation of the RDF language – which could be more effective in actual applications –, the only normative one is the model-theoretic semantic provided in the just cited document, where it is also required a monotonic discipline on the language (and its interpretation) to correctly generate inferences (which is essentially entailment). Together with these acknowledgements there is also the one that admits the limitations of such a semantic – that is to say: no illusion is feeded about Semantic Web ability to deal with the real, full meaning[16] of data and information, since the semantics of SW languages cannot be but formal semantic theories, with pros and cons that formal logics know and show. Then it is a little surprising to find two uncanonical concessions that RDF allows. The first is that RDF lets to produce infinitely descending chains of membership, which violate the foudation axiom of standard Zermelo-Fraenkel set theory. In particular, RDF let properties apply to themselves, and classes to contain themselves. But in order to recover compliance with set theory, RDF exploits a mapping mechanism that distinguishes the name of a class or property from its extension, and also causes the second uncanonical aspect, that is its ambivalence on the extensional and intensional level. This ambivalent behaviour introduced with RDF deserves more attention at next level language, where it causes the separation of two distinct "evolution branches".

Provisional morals that we can derive up to now may be that set theory does not fit adequately Semantic Web needs.

1.2.3. Representing knowledge with owl

What we have called distinct evolution branches of RDF are the two main variants of the Web Ontology Language (OWL), built on top of RDF and released by W3C in 2004. OWL in fact offers more than two variants (or profiles) – they are currently six – but we may clearly distinguish between the only variant that enriches and improves RDF expressive power still keeping all RDF theoretical assumptions, that is in particular the issues about its semantics, and all the other profiles that "deactivate" them in

[16] It is well known that the meaning of a term, of a sentence, or of any piece of information, intended as what a human can understand, is the result of many factors among which we may count e.g. culture, social conventions, personal history . . . which reach far beyond what a formal language can catch.

order to guarantee decidability (and optimize tractability) for automatic reasoning applications that process OWL. Just a little of history to explain that: in the time between the first release of RDF and the publication of OWL Semantic Web community had strenghten its relations with that of knowledge engineering and felt the need to prepare Semantic Web in order to let reasoners (theorem provers) integrate in it and act an important role for the check, use and discovery of the knowledge sparse in the Web as pieces of information. From a logical point of view, an RDF or OWL vocabulary (possibly together with a collection of resource annotations) forms a theory. Such a theory could not be properly managed by a reasoner in the case of RDF (and of OWL-Full) due to the uncanonical aspects of their semantic interpretation. But with all the other OWL profiles this problem does not hold. Quite the contrary, each of them can be put in a neat correspondence with a special logical formalism from the family of Description Logics (DLs) which, on the one hand, offer an alternative semantic interpretation with respect to the original one for RDF, and on the other hand make these OWL profiles ready-to-use with theorem provers, since DLs are studied precisely to be well tractable with reasoners and in turn reasoners undergo unending optimization to increase performances in processing the different DLs. OWL-DL and OWL-Lite come from the first version of OWL (the one dated 2004) and implement two different DLs, the former's one being more expressive than the latter's – that is OWL-DL offers some more operators (or class constructor) to express and define classes and properties in OWL ontologies (as are to be called OWL vocabularies). The last profile of the first OWL version is precisely the undecidable OWL-Full, which uses all the same operators than OWL-DL but has no restrictions on their use: again a property can apply to itself, or a universal class exist that contains any other class including itself, or even two classes have exactly the same extension and still be two different classes, since their diversity can be stated at intensional level. Three other OWL profiles come from the more recent OWL2 specification (which dates 2009). It considers a set of new operators and re-assort the previous ones with the new ones in order to provide the profiles OWL2-EL, OWL2-QL and OWL2-RL which differ not only for their expressive power but also for the purpose they are intended to fulfill. They come from a very fine work of optimisation with respect to specific reasoning and information retrieval tasks so that, for instance, OWL2-QL offers its support to conjunctive query answering, while OWL2-RL introduces rules firing into ontologies.

Precisely these two areas (rules processing and query answering) are the other main interests at this level of the Semantic Web stack, along with ontology. As regards queries, that is punctual information retrieval from within a collection of OWL or RDF annotated resources (a datastore), W3C has released a standard language named SPARQL about which we will talk longer in the following (see 2.6.2). As regards rules, although it is a quite simple and traditional way in computer science to encode knowledge, there is not yet an official W3C recommendation; the group of experts in charge of a standard for rules processing over Semantic Web is still working on the task. The main interest anyway will be posed on an exchange format to let already existing rule systems (and languages to express rules) to fire their rules over the Web – the name of the resulting specification should be Rule Interchange Format (RIF).

More details about characteristics of the ontologies that can be designed with OWL are given in the following chapter 2.

1.2.4. Other upper levels.

Next levels in the SW stack, called "unifying logic" and "proof" – just over OWL, SPARQL and RIF – still stays quite mysterious: as the picture (1.1) shows, there is no recommendation nor active working group in charge of developping a standard way to reason on (i.e. to use according to some logical system) the logically strcutured information that nowadays already populate Semantic Web, or to verify the correctness of an inference. Yet we said above that our interest about Semantic Web is mostly at this level. It is precisely because there is nothing well defined as regards this part of Semantic Web that we may think to offer our little contribution to the understanding of what could be there.

An interesting document hosted at W3C website [Berners-Lee, 1999-2009], although purely informal, offers some enlightening statements about what Berners-Lee himself expects these levels to be. We can summarize his (admittedly still unripe) position in the following points.

- A first condition envisaged for the "logical exploitation" of Semantic Web is that "it must have a well defined semantics so that one can say precisely what is being represented". No need to say that here he's speaking of formal semantics.
- Since SW would relie on both expressive languages to represent human knowledge and an "efficient, powerful, and understandable" reasoning mechanism, but there is a clear trade-off between expressive power and tractability (even decidability), expectations about reasoning may be sacrificed in favor of expressivity.
- There is no real intention (neither real need) to have one (unique) reasoning system to process Semantic Web knowledge. A feasible scenario for reasoning within Semantic Web is sketched based on reduction of what could be genuine Knowledge Representation expectations. In Berners-Lee's words, "semantic web itself will not define a reasoning engine. It will define valid operations, and will require consistency for them. On the semantic web in general, a party must be able to follow a proof of a theorem but is not expected to generate one".
- Finally, reasoning on Semantic Web should become a local and context-aware process, executed by a large variety of reasoners which may use any system-specific language. The only inescapable point must be the ability to produce and expose an output that respects minimal commitments of Semantic Web (e.g. the equivalence with the standard semantics of RDF) and in a format that can be managed by any other existing reasoning system, so that new, inferred knowledge get publicly shared as additional source of information.

The first point is matched with RDF and OWL semantics. However, the same semantics contributes to raise problems at the second point, which in turn is partially matched with the different versions and profiles of OWL. The second point turns out also to be somehow linked to the secondary issue about human readability of SW vocabularies or ontologies: if machine understandability is kept under special conditions, and ontologies are welcome to express knowledge unmanageable by inference systems, for any other use of the information and knowledge put within such documents that is not encoded in *ad hoc* programming, it needs human intervention. We will come again on this point while discussing deeper on ontologies. The third and fourth points are to be read together so as to realize that reasoning is not, strictly speaking, a matter at the core of Semantic Web interests. Indeed we find that the "unifying logic" layer is expected to eventually work in a fashion very similar to the proposed RIF format for rules exchange, with the difference that it regards any kind of inference drawing. Once in service, such layer should therefore allow to share the results of theorem proving – it does not matter who and how makes it – about knowledge bases which are not necessarily connected among them. In the same direction then, the next level, called "proof" has to guarantee nothing but the ability to show the inference steps (indicate all the premises used, maybe also locate them on the Web, we could say: contextualize the result already proved by someone else) and certify the correctness of the conclusion drawn, in order to justify the behaviour of a web agent.

However, since there is no other relevant, more detailed, most of all official pronunciation about what have to be SW logic and proof layers, for the time being we can just retain these few basic ideas – which are very unlikely to change in the current state of Web and Semantic Web – and try to respond them in an original, logically interesting and as complete[17] as possible way.

[17] Where for completeness here we mean the ability to account for every logical interaction that may happen in the Web on logically structured data, information, knowledge.

2. Ontologies

Ontologies are the top level of data and information description, so that one may really talk about knowledge: the intellectual form of structured information. Semantic Web ontologies are the result of the merge of the long experience of Artificial Intelligence research for models with which formally represent knowledge – in order to transfer it to machines – together with the Web. Such technical-scientific operation, on the one hand, brings AI in the pretty new (for it) field of a really open context like the Web. That means in particular that it gets out of the unrealistic scenario of a single intelligent agent that knows all about the world that it has to deal with, and, most of all, where it is the only active agent. In our opinion in fact this is the most important return that AI gets from its recycling in the World Wide Web. On the other hand, the merge provides Semantic Web with a number of results coming from quite a long history – at least, much longer than the Web history – which promises to help Semantic Web to get in service.

2.1. Principles of ontologies

Out of other models to represent knowledge all the same developed and used within AI such as semantic networks, frames and rules – sometimes even with very successful results as it is the case of rules and rule engines which are main components of expert systems – the one chosen to construct the core of Semantic Web knowledge is that of ontologies[1].

The choice is definitely reasonable if one considers what is below the OWL level in SW stack, i.e. vocabularies and "lexica". Indeed a SW ontology can easily be seen as the richest form of vocabulary, where along with descriptions of terms one finds also real formal definitions, so that specific relations among the terms get more significance and logical meaning. In the same time, however, the enrichment causes also some complications as regards the status of such a knowledge description. Since it gives logical definitions, since it refers plainly to the concepts beside a term, since it allows to express quite complex axioms, it also commits on really ontological issues. In fact, also our reduction of ontology to "the richest form of vocabulary" is to be taken as meaningful only within Semantic Web and Knowledge Representation small worlds.

[1] It is only quite recently that W3C felt the need to reserve some room within Semantic Web architecture to rules, after having acknowledged the insatisfaction of knowledge engineers for being not able to express in ontologies some (even conceptually very simple) forms of knowledge. Nevertheless, independently from the current absence of any standard about SW rules, ontologies are not going to be replaced in their role of cornerstones of Semantic Web – mind for instance the ancillary role that a mere interchange format, like RIF is to be, may assign to rules.

But also within these small worlds, an ontology raises a number of deep and sometimes controversial issues.

Let's try to shed some light on these points by starting with a better characterization of such ontologies. Here the most known definition of Semantic Web ontologies (by Gruber), which nevertheless is often found to be controversial [Gruber, 1993]:

> An ontology is an explicit specification of a (shared) conceptualization.

The adjective *shared* is put in brackets since it really appears only in a later version of the definition, given by Borst in 1997 [Borst, 1997], so that the version missing it is somehow better known. One may feel immediately the distance with respect to traditional philosophical discipline of ontology – the one strictly intertwined with metaphysics – as words like "being", or "exist" do not even appear in this definition of ontologies[2]. Therefore it may look like an abuse to call ontologies such "just a little more than formalized vocabularies", but we will find arguments enough to justify (or at least make it reasonable) that name.

Being so concise, this definition needs further explication before we pose any appreciation or criticism; we will provide explication firstly by proposing a second definition (by Studer et al. [Studer et al., 1998]) of ontologies for Knowledge Representation that plainly recovers the previous one and deepens and clarifies it:

> An ontology is a formal, explicit specification of a shared conceptualisation. A "conceptualisation" refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. "Explicit" means that the type of concepts used, and the constraints on their use are explicitly defined. For example, in medical domains, the concepts are diseases and symptoms, the relations between them are causal and a constraint is that a disease cannot cause itself. "Formal" refers to the fact that the ontology should be machine readable, which excludes natural language. "Shared" reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.

First of all we spend a few words to clarify the technical meaning of terms like *concept* and *relation*. Such terms indeed are to be read in continuity with XML and RDF experience. That is a concept is the idea behind a term like those in RDF vocabularies (or the name of an element in an XML schema). It may sound quite puzzling that a formal model to represent knowledge in such a way that machines can use that knowledge cares about ideas. But in fact one speaks of concepts in ontologies just to put together two main aspects of the meaning of a term: its intension and its extension.

[2]The reader may have noticed the distinct use of the singular and plural form of the word *ontology*: when it is the philosophical discipline it is in the singular form, whilst the plural form stands for Artificial Intelligence knowledge representations. It may happen also to talk about just one particular KR ontology, or about the general model of ontology for KR; to distinguish even in this case it has been proposed [Guarino, 1995] a distinct use also of the capital letter, so that philosophical discipline deserves the uppercase O – at least for its millenary history – whilst KR ontology writes as "lowercase ontology". We prefer not to adopt such a convenience since it will be always clear enough which kind of ontology we will deal with at any time.

According to a long tradition that goes back up to Aristotle, and that is well rooted in the area of Knowledge Representation, meaning is a triadic relation among symbols of a language, objects in the world and ideas, or concepts. Then, to represent knowledge (in KE sense) is basically to instruct a machine to deal with symbols, terms in such a way to act on the objects which are their denotation, their reference into (some kind of) reality, in such a way as if the machine acknowledges also the ideas, the concepts that are behind the symbols and that allow to keep together a class of objects all responding to a same name – i.e. to act *as if the machine knows what those objects are*.

So a concept in a SW ontology is the intensional counterpart of a class of objects (usually called individuals or instances) that can be grouped together under a common name, the term that "locates" the concept within an ontology. Sure, the concept may be finely depicted thanks to discoursive descriptions within the ontology and, most of all, it can be somehow delimited in the form of a logical structure that poses it in relation with other (concept-denoting) terms; but in no occasion the very concept is grasped (nor it is really intended to be) by the machines, or the "intelligent" agents that relies on ontologies to execute some task in the Web.

Also relations are seen in the same dual perspective so that, along with a special term to denote each relation, and possibly a discoursive description to explain how it is to be intended the relation, one is given the extension of the relation as a subset of the cartesian product of the two[3] classes of objects (we would say concepts, but the cartesian product of two concepts may sound a little unusual) which are assigned as domain and range of the relation. In practice, the extension of a relation is the collection of pairs of objects (Web resources for example) that (are known to) exemplify that relation. The formal definition of a relation in an ontology indeed is mostly this: the choice of its domain and range sets, plus some optional features depending on the profile of the language chosen (for instance, in OWL-DL a relation may be functional – i.e. it is a function –, inverse functional, symmetric and transitive).

The double perspective on reality (intension and extension) is structurally realized in KR ontologies, and then also in SW ontologies, through the possibility to consider an ontology (and often also actually produce and use it) as made of two different parts, called boxes: the T-box and the A-box, which together form a Knowledge Base (KB). In the T-box (box of Terminology) are represented the intensional aspects of a domain: terms to name concepts and relations are introduced and defined here; while the A-box (box of Assertions) stores the collections of individuals and pairs of individuals that populate, instantiate concepts and relations respectively.

Having made a little more precise our vocabulary about ontologies, we may better comment on the proposed definitions. Since the second is a larger exposition of the first one we will skip the first, but it was relevant to mention it since it has been a sort of flag for years for SW "ontologists", apart from the fact that it is "introductory" to the second one.

The first step to have an ontology is then, according to Studer et al., to produce

[3]There can be only binary relations in Semantic Web ontologies due to OWL, and RDF before it, design and foundational limitations – remember of the basic form of information storing which is a binary predicate.

a conceptualization. Before we question what is to be a conceptualization, we want to ask what the conceptualization should be about? Indeed, philosophical tradition of ontology would keep the broadest horizon as its own field of inquiry. On the contrary, that definition talks us about “some phenomenon” in the world as the subject of interest for a conceptualization. However it should now be unsurprising: SW ontologies – and all the same Artificial Intelligence ontologies – are designed for some specific, well defined and delimited purpose, and not for the sake of science or knowledge itself. Therefore, there is no interest in SW for ontologies about most general and critical issues such as being, existence, reality, ethics, God, the universe and all the rest, but only for relatively small, simple and quiet portions of reality such as for instance industrial machinery, bacteria, business processes, genes, diseases. . . To be honest, this is entirely true except for the case of *foundational* ontologies, which try to provide a philosophically aware account of fundamental concepts – e.g. defining what is an object, a process, a property, sometimes also what space and time are – in order to alleviate the task for specific domain ontologists who may then design ontologies without troubling with most basic and “difficult” concepts. Nevertheless, most of SW ontologies are concerned with some particular restricted area of interest, usually known as a *domain*, so that an ontology designed to represent any of them is called domain ontology, whilst an *application* ontology is an even narrower scope ontology whose domain is strictly the minimal part of reality that some software program needs to “know” in order to execute the tasks it is written for.

As regards the conceptualization, this is the most delicate part of ontology design – by the way the main criticism to Gruber’s definition is appointed precisely on the notion of conceptualization. Technically speaking the conceptualization for a SW ontology is what allows to identify concepts, and (binary[4]) relations among concepts, which are relevant to the domain of interest. It is quite easy to guess why the most of issues about ontologies are involved with such conceptualizations. However, since they would deserve careful discussion in a setting that accounts also for fundamental acquisitions in the tradition of philosophy – after all it is ontology at the core – we will highlight them later in this chapter and then we will discuss them deeply and largely only in the final part of this work which is devoted precisely to the philosophical aspects of Semantic Web, ontologies and the challenges that they pose to contemporary philosophy. Here we concentrates just on a couple of them, namely reliability and generality[5], as they clearly give the flavour of the operational and instrumental nature of SW ontologies. Indeed any two individuals, even with the same background, may conceptualize in different fashions the same domain; who is then entitled to provide the conceptualizations that are the core of the knowledge of an “intelligent” system in the Web which should then be available for consultation from anywhere in the world? Like in the tradition of AI, usually the authors of conceptualizations are domain ex-

[4] To deal with predicates whose arity is bigger than 2 requires a bit of creativity to find some viable workaround, like for instance decomposing the relation in many binary subrelations. Or the alternative is surrender to language constraints.

[5] For reliability we mean the adequacy of an ontology to correctly describe some domain; for generality, the possibility for the ontology to be accepted by a community of users as large as possible, in principle everybody.

perts (i.e. scholars or scientists according to the particular domain), which are helped in their task by one or more knoweldge engineers. But the proficiency of a scholar or scientist may not be a sufficient warranty. This makes another point of Gruber's and Studer's definitions: the conceptualization has to be shared, that is the result of a negotiation among experts, scientists and not a private affair – what should assure a greater level of generality. However no definition may decide a precise extent to which the conceptualization should be shared, nor the level of proficiency of the domain experts, so that reliability and generality of SW ontologies depends just on the good intentions of people that engage the task of building an ontology.

Once a shared conceptualization of the domain of interest is achieved, we obtain a KR / SW ontology by means of explicit formalization of the conceptualization – which to us sounds quite an oxymoron. Indeed, on the one hand the request for formality, better: formalism in this case, is exactly what enables computer programs to perform complex tasks in the Web by "understanding" the meaning of the information that they find and handle in web pages or in public data sources. Since all the meaning that programs can grasp is in terms of logical relations among types (of objects, of data), we must encode the knowledge in the best, richest and most suitable formal language at our disposal so as to save the most of what we can say about it in logical terms. Then, on the other hand, it sounds a little surprising to read in Studer's definition that explicitness is plain explanation of the particular objects and relations like, in their example, diseases, symptoms and a causal relation from the former to the latter. Diseases, symptoms and causality will stay always beyond any program's understanding; what it could retain from such a representation is just that there is a particular set of pairs denoted by a name like `causes` whose first element must be also an element of the set of diseases and the second of that of symptoms.

Also the subsequent passage "'Formal' refers to the fact that the ontology should be machine readable, which excludes natural language", should be confronted with the fact that the exclusion of natural language concerns the formalism with which the global description is to be presented, and is not a radical refuse – that is in particular we still have the naming of concepts and relations by means of their (more or less scientific) names in some particular natural language, which is very likely to be English. Nevertheless, there is no reason why machine could prefer entities named with their natural language names with respect to any other arbitrary string. The reason why natural language names are actually used bring us back again to the same point: human readability is not a marginal interest in designing ontologies, which definitely are not really conceived only for machines.

To mind human readability, although within a formalized language, means to guarantee ease for the human expert that has to supervise the proper working of the system to perform some tasks which are crucial in a knowledge based system: to check the overall consistency (in the sense of being meaningful with respect to the external reality) of the operations performed by the system; to intervene on the ontology in order to fix and update the knowledge stored in it, for example in the case of discovery of a "conceptual bug", or a new acquisition in the scientific domain represented in the ontology requires to change some axiom, some terms or anything else – this point incidentally causes us to mention the issue of non-monotonicity, that we will face in more detail in the follwing; and, finally and most of all, to support processes of comparison and integration between different ontologies. To compare ontologies indeed requires, much more than a comparison of the formalisms in which they are presented, to understand the point of view according to which the domain is described.

Yes, we admit, these remarks may look quite like subtleties, but they are illuminating about the (many-times-)mentioned issue of human readability and let us eventually close the discussion on that point. Many efforts in designing good ontologies in fact would be nonsensical if all the ontology should be only read and used by special programs, think for example of the good practice of annotating with rich descriptions any concept and relation or the choice of the most appropriate terms to denote them. The real advantage of well designed SW ontologies is then the possibility to "recycle" them, i.e. reuse for other applications, different from the original one for which they are conceived. But to get this recycling process properly working it is necessary that the original intended meaning is preserved – in such a way to preserve global coherence and, from a more pragmatic point of view, the compatibility of the different services that will use the same ontology and could also exchange data with each other. While the original application, being designed together with the ontology, surely respects the ideas and assumptions that domain experts put behind any concept and relation of their conceptualization, any other application – if it were only a matter of logical meaning – could obey axioms and restrictions encoded in the formal reading of the ontology yet deeply misunderstanding the real meaning of the information that it manipulates. So, at any time an ontology is to be read outside the original setting it is designed for, it needs the careful understanding of a human being to be still useful, for instance in order to be properly integrated in a new, different system.

In conclusion, the morals that we can draw as regards the *ratio* between formalism and explicitness is simply that the more is said, in logical form, the more room for inference drawing there will be. So explicitness in Studer's example is not a plain explanation of (clinical) causality, but the imposition of domain and range restrictions on the relation that expresses causality, so as to prevent situations like symptoms asserted to cause some disease – even though a medical doctor could have something to say on that point.

2.2. Actuality of ontologies

Nowadays we can find a number of ontologies in the (Semantic) Web. Most, mainly the largest ones, are related to important research projects conducted by great organizations and research centres. These large ontologies may be designed either in order to help in the information and knowledge sharing processes among scientists of different branches and all over the world – so that the ontology is devoted to specific needs of those communities and its possible contribution to Semantic Web is rather accidental, the ontology being often not even (fully) available in a W3C standard language (e.g.

GO[6], GALEN[7], SWEET[8], ...) – or primarily in order to contribute to Semantic Web building (for instance DOLCE[9] and BFO[10]). Beside such large ontologies (sometimes colossal ontologies inasmuch as they may count even thousands of concepts) we find also ontologies which are the result of a re-presentation of other sets of metadata arranged to be Semantic Web-friendly *via* a translation in OWL or RDF (e.g. the Dublin Core[11]). Other ontologies have been designed by (quite small) communities to provide some interesting and very simple tools to start actively working on Semantic Web by stimulating general user (Web contributors) to adopt such ontologies and annotate accordingly their documents (maybe the best known example is FOAF[12]). Finally, a number of ontologies are produced and used within the restricted areas of information systems in big companies, building what is called corporate Semantic Web, that is a Semantic Web within an intranet where all the knowledge assets of a company are structured and shared among workers and heads of service in order to improve problem solving as well as business solutions. These last ontologies are somewhat hidden, but nonetheless they are the ones closest to realize the ideals of Semantic Web, since they are actually used with noticeable advantages for companies. It is easy to guess that the restricted domain of a company's interests and activities, though it may be even a huge multinational company, together with an easily identifiable point of view on (business) reality, greatly facilitate the task of getting the knowledge based system in service and largely used by workers. The drawback of these ontologies is that they are out of the free information exchange of the Web.

Beside this summary classification of ontologies by their purpose, we may distinguish also some different approaches to ontology building that characterize, along with the

[6] GO (Gene Ontology) is a very large bioinformatics project started in 1999 which aims to standardize how genes attributes, and genes products, should be represented in a species-independent manner so as to allow for cross-interrogation of different (large and important) gene databases. Although it is not its "natural" language, there exists and is constantly mantained an OWL version – `http://www.geneontology.org`.

[7] GALEN (Generalized Architecture for Languages, Encyclopaedias and Nomenclatures in medicine) is a EU-funded project started in 1992 for the harmonization of clinical terminologies in order to favour exchange and re-use of medical information; the resulting ontology is available also in OWL – `http://www.openclinical.org/prj_galen.html`.

[8] SWEET (Semantic Web for Earth and Environmental Terminology) is a collection of ontologies developed by the Jet Propulsion Laboratory (NASA) ranging from the description of the elements of many Earth system sciences to the representation of measurement systems (time and space) – `http://sweet.jpl.nasa.gov/ontology`.

[9] DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) is a genuine foundational ontology, developed within a EU-funded project as a module of a larger collection. It is originally written (on paper) in first order logic with modalities; what could be expressed in DLs has been then implemented in OWL – `http://dolce.semanticweb.org`.

[10] BFO (Basic Formal Ontology) is a foundational ontology especially conceived to support other ontologies developed in the medical and biomedical area. As with many biomedical ontologies, OWL is not the first choice language for implementation, but there is a suitable translation – `http://www.ifomis.org/bfo`.

[11] The Dublin Core Metadata Element Set is the result of an enduring work started in 1995 to define a set of metadata to describe any digital resource accessible *via* computer networks. It has been readily translated in OWL to be properly used for Semantic Web – `http://dublincore.org/documents/dces`.

[12] FOAF (Friend Of A Friend), further details are given in next section – `http://www.foaf-project.org`.

different uses, the main kinds of Semantic Web ontologies. Please note that here we aim to signal only main characteristics and differences among many methodologies for ontology design that have been conceived and refined in about twenty years of work on ontologies; we are not going into a full discussion on this point. For an in-depth discussion we suggest [Noy and Hafner, 1997] and [Lembo et al., 2006].

Ontologies of the first type are developed with a design process that takes even years of "negotiations" among a relatively large team of domain experts who produce a fine conceptualization based on a primarily theoretical account of the domain. This is the prototypical top-down approach, which eventually delivers a high quality ontology, made only of a (very large) T-box, which can be used in two ways: as the core of a collection of services and applications to be helpful for some particular scientific communities; or as a foundational ontology, i.e. an ontology offered to the SW community to be used, through the "import mechanism" (nothing more than the ability to recall a name space *via* its URI), as a reference to give a richer frame to domain specifc concepts defined in smaller ontologies. Besides the possibility to facilitate the design of domain ontologies, foundational (or upper) ontologies have been thought also as a viable way to give coherence to Semantic Web. Indeed, if any single ontology, designed by no matter who, is linked to some foundational ontology, and foundational ontologies also are linked with each other, we (and our computer programs) will be able to understand any (SW) concept in relation with any other concept. Such a road to World Wide Semantic Web does not meet any major technical obstacle, but it clashes against major conceptual, cultural and philosophical problems as we will discuss in next section.

On the contrary, the last kind of ontologies – be they precisely for corporate SW or, generally speaking, domain specific ontologies for small applied research projects or even business to customer services – they are all built on the basis of a large exploitation of "semantic tools" which provide a first, rough terminology which is subsequently reviewed by domain experts in a process that should not take too long – since it is to realize to some extent a business (or research) utility – and which delivers eventually a T-box whose conceptual quality may even be not very high, but it comes along with a large A-box (the set of dcumental resources, or data), so that such ontologies become actually usable for enhanced information retrieval. For "semantic tools", that we have just mentioned, we mean a large variety of techniques that help knowledge engineers, but also domain experts, to identify the concepts that deserve a mention in the ontology, thus performing a sort of computer aided conceptualization. Such tools exploit state of the art data and text mining techniques in order to identify candidate concepts, typically thanks to the analysis of databases and/or of large *corpora* of documents relevant to the domain of interest. It is obviously a stochastic approach to Knowledge Representation based on linguistic analysis (most frequent terms, recurring expressions, and weighting *criteria* to detect also some hierarchy among candidate concepts) often also with a contribution from machine learning. It invariably requires the refinement of human experts to discard fake-concepts and then to give also good descriptions and possibly definitions of the concepts. By the way, it is significant that the design and development of algorithms to accomplish such tasks is one of the most active research areas related to Semantic Web. In such a methodology, after all, is usually recognized a bottom-up approach to Knowledge Representation, even though in our opinion it is bottom-up only as far as it is opposed to the classical top-down approach.

In the middle between these two approaches there are: i) the one that we would call the "translational approach", that is to take a pre-existing metadata schema and convert it into a SW standard language – which offers the advantage that the conceptualization has already been made so that it preserves the original quality, whichever it is, and the resulting ontology is soon ready to describe collections of resources publicly available in the Web; ii) and the community driven ontology building, which really should be differentiated in many different procedures. All of them however basically rely on a relatively small community of SW researchers (or even fans) rather than domain experts, who want to substantiate SW ideas. Thus, they design ontologies where the re-cycling of other ontologies (by referring to concepts defined elsewhere) is very frequent so as to get both their own concepts framed in bigger ontologies (and finer logical and conceptual structures) and also SW ontologies interconnected. Then they provide resources described according to the concepts and relations that they have defined, and promote their ontology by putting in circulation their annotated documents and stimulating other people to do the same, like in social networking. Besides the FOAF experience there are a number of other vocabularies (rather than regular ontologies) defined in special formats[13] which may not fully comply with W3C standards but which are in use in the Web. Typically they all propose some kind of formal specification of quite simple information, like in most cases personal (contact) information and friendship or acquaintance relations.

After so many words, we are finally going to show some code, that is an XML/RDF serialization of a Semantic Web ontology. It is not easy to choose which ontology to adopt to give examples: ontologies from the first type are often huge in T-box but do not have any A-box so that we have no opportunity to show how data can be described by an ontology. We can not even produce a fictional A-box since it would make no sense – what could it be the data for a concept like `spatio-temporal region`[14]? Such ontologies indeed are made to conceptually empower the less philosophically committed domain ontologies. Ontologies from the second type, on the contrary, have interesting A-boxes – collections of data, even very large, which are described by the corresponding T-box – but often T-boxes of no interest: few logically poor defintions and too short term descriptions, when available. Thus our choice has gone on the last kind of ontologies, which are also interesting for they are the result of somewhat spontaneous initiatives. The complexity of T-box axioms really is not so important not even in these ontologies, but such ontologies largely recall other, more logically complex ontologies, thus inheriting fine logical meaning for their own concepts, and at the same time also allow to produce meaningful annotations of documents and resources, or data labelling – that is marking data to be compliant with the object types defined in the ontology, that are the concepts.

[13] For instance the whole family of microformats, which are basically metadata sets defined in simple (X)HTML.

[14] From DOLCE ontology.

Below there is an excerpt[15] from the RDF/OWL version of the FOAF (Friend Of A Friend) vocabulary, presented in XML-style syntax. As you may note, we call it a vocabulary, since within the FOAF project it is officially called like that; nevertheless, we have explained above the tight relation between vocabularies and ontologies in Semantic Web, for which they basically differ for the language they are written in, so that ontologies are somewhat a richer form of vocabulary thanks to the use of more expressive languages which allow to produce more complex term/concept definitions. Such ambiguity about the status of vocabularies and ontologies in Semantic Web is made apparent even in the specification here below, where one can read, thanks to visible comments in the code, that is called a vocabulary but is formally defined as an ontology in order to exploit capabilities of OWL.

2.3. Example – Excerpts from FOAF ontology.

The FOAF project aims to enrich the Web with simple – hence easy to spread largely over the net – semantic annotations concerning personal identity, people's contacts, work activities and similar things. It makes large use of W3C's standards for Semantic Web and also relies on other ontologies to get its own concepts interestingly framed into finer conceptualizations. Here below some lines from the original FOAF specification:

```
<!-- This is the FOAF formal vocabulary description,
   expressed using W3C RDFS and OWL markup. foaf/spec version -->
<!-- first we introduce a number of RDF namespaces we will be using... -->
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:vs="http://www.w3.org/2003/06/sw-vocab-status/ns#"
   xmlns:foaf="http://xmlns.com/foaf/0.1/"
   xmlns:wot="http://xmlns.com/wot/0.1/"
   xmlns:dc="http://purl.org/dc/elements/1.1/">

<!-- Here we describe general characteristics
   of the FOAF vocabulary ('ontology'). -->
<owl:Ontology rdf:about="http://xmlns.com/foaf/0.1/"
   dc:title="Friend of a Friend (FOAF) vocabulary"
   dc:description="The Friend of a Friend (FOAF) RDF vocabulary,
      described using W3C RDF Schema and the Web Ontology Language.">
</owl:Ontology>

<!-- FOAF classes (types) are listed first. -->
<rdfs:Class rdf:about="http://xmlns.com/foaf/0.1/Person"
      rdfs:label="Person" rdfs:comment="A person.">
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
   <rdfs:subClassOf>
      <owl:Class rdf:about="http://xmlns.com/foaf/0.1/Agent"/>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
   <owl:Class rdf:about="http://www.w3.org/2000/10/swap/pim/contact#Person"
      rdfs:label="Person"/>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
```

[15] For the full ontology please refer to `http://xmlns.com/foaf/spec`.

```
      <owl:Class rdf:about="http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing"
      rdfs:label="Spatial Thing"/></rdfs:subClassOf>
    <!-- aside: are spatial things always spatially located?
      Person includes imaginary people... discuss... -->
   <rdfs:isDefinedBy rdf:resource="http://xmlns.com/foaf/0.1/"/>
   <owl:disjointWith rdf:resource="http://xmlns.com/foaf/0.1/Document"/>
   <owl:disjointWith rdf:resource="http://xmlns.com/foaf/0.1/Organization"/>
   <owl:disjointWith rdf:resource="http://xmlns.com/foaf/0.1/Project"/>
</rdfs:Class>

[...]

<!-- FOAF properties (ie. relationships). -->
<rdf:Property rdf:about="http://xmlns.com/foaf/0.1/mbox"
      rdfs:label="personal mailbox" rdfs:comment="A personal mailbox,
      ie. an Internet mailbox associated with exactly one owner, the first
      owner of this mailbox. This is a 'static inverse functional property',
      in that there is (across time and change) at most one individual that
      ever has any particular value for foaf:mbox.">
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
   <rdfs:domain rdf:resource="http://xmlns.com/foaf/0.1/Agent"/>
   <rdfs:range rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
   <rdfs:isDefinedBy rdf:resource="http://xmlns.com/foaf/0.1/"/>
</rdf:Property>

[...]

</rdf:RDF>
```

The first block of the code, after some comments, is a preamble where the name spaces that will be recalled frequently in the rest of the ontology are introduced and given a prefix to ease referencing. In the second block, the ontology gets its own namespace and some metadata about itself: they are just a few attributes from the Dublin Core metadata set. The third block is about class/concepts definitions. To give just an example we have chosen to show only one concept: `Person`, that is the concept of person in FOAF; according to the specification given, it is to be understood as a subconcept (subtype, subclass) of the concept `Person` as it is defined in the vocabulary/ontology whose namespace is `http://www.w3.org/2000/10/swap/pim/contact` – it is a similar purpose ontology developed by Berners-Lee among others – and also as a subconcept of the concept `SpatialThing` from `http://www.w3.org/2003/01/geo/wgs84_pos` which in turn is a completely different purpose ontology (it is about geodetic referencing) but once again counts Berners-Lee among its authors. Such a mixture of apparently distant concepts reveals something of the real nature of Semantic Web where people can link any two pieces of knowledge, and a semantic web agent can follow them acquiring new information at every step – with the conviction (or the hope) that after a number of passages everything gets clear. Besides its subclass relations, to the definition of the concept `Person` contribute also a series of disjointness axiom which involve other concepts in the same ontology. That is to say that `Person` is a distinct concept with respect to a `Document`, an `Organization` or a `Project`. The last block is about relations (OWL and RDF properties). Once again we quote only one just to give the idea. Besides the assignment of domain and range classes, it is noteworthy the presence of a discoursive description of the relation that allow a human reader to fully understand what is to mean an `mbox` attribute according to FOAF.

Figure 2.1 is to give a better idea of what the FOAF ontology is about, although it shows only the class hierarchy, that is the concepts without their specific relations (except for the IS-A, or subsumption, relation which underpins the hierarchy).

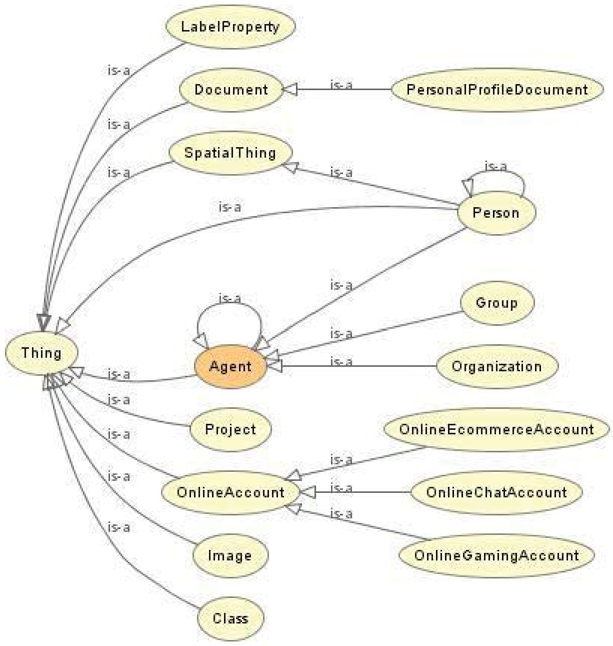


Figure 2.1.: FOAF concepts.

Whereas the code above shows some pieces of the FOAF vocabulary but says nothing about actual data described according the ontology – that is an (all the same partial) A-box – here follows just a simple example of what looks like an RDF annotation of a document compliant with the FOAF vocabulary:

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/">
<foaf:Person rdf:ID="me">
    <foaf:name>Marco Romano</foaf:name>
    <foaf:workInfoHomepage rdf:resource="http://logica.uniroma3.it/~romano"/>
    <foaf:mbox rdf:resource="mailto:m.romano@uniroma3.it"/>
    <foaf:knows>
        <foaf:Person>
            <foaf:name>V. Michele Abrusci</foaf:name>
            <foaf:mbox rdf:resource="mailto:abrusci@uniroma3.it"/>
        </foaf:Person>
    </foaf:knows>
```

```
    <foaf:knows>
        <foaf:Person>
            <foaf:name>Christophe Fouquere</foaf:name>
            <foaf:mbox rdf:resource="mailto:cf@lipn.univ-paris13.fr"/>
        </foaf:Person>
    </foaf:knows>
</foaf:Person>
</rdf:RDF>
```

2.4. Meaning and semantics

As far as the term *ontology* is used with so many and subtly different acceptations in the fields of information systems, Artificial Intelligence and Semantic Web, it is a hard task to find a definition that satisfies everybody. Just to give an idea of the variety of different acceptations that the term ontology may assume still in the field of computer science – thus excluding the whole philosphical area – we propose a very explicit picture from Smith ([Smith and Welty, 2001]) which shows in a glance the variety of “things” that have been found (in a research by [Welty et al., 1999]) to be referred to, at least at some time, as ontologies.

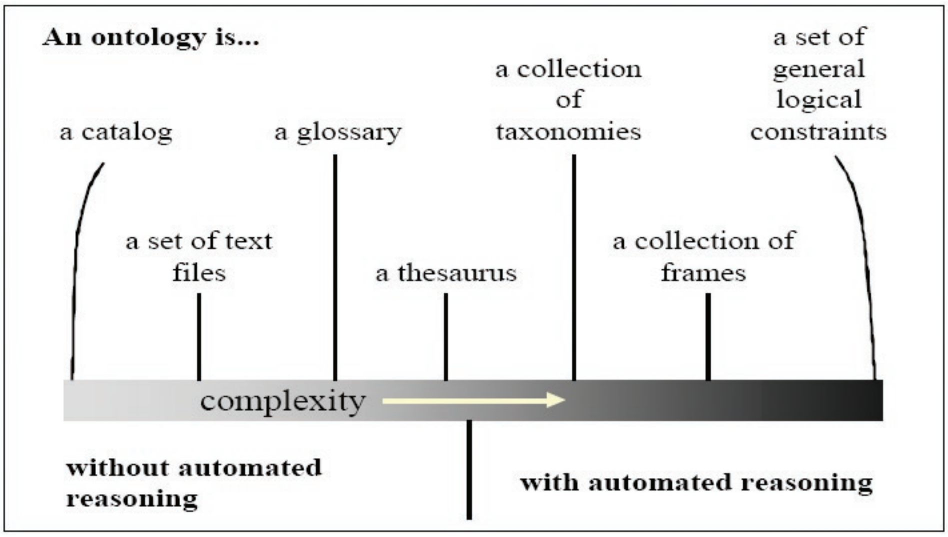


Figure 2.2.: The wide range of ontologies

As a consequence of such a variety of uses, and of the apparent confusion about what is to be essentially an ontology, even the largely accepted definition by Gruber has received some criticisms over time. We too have attempted in last pages to subtly refine the definitions by Gruber and Studer in such a way as to specially restrict the discourse to ontologies for the Semantic Web, which are our primary object of interest for this work, even though we have proposed no additional alternative definition. However, we still have to make clear some aspects concerning the notions of *meaning* and *semantics*, and the capability of language and logic to account for them. For this purpose it is useful to briefly recover a part of the debate on the definition of ontology.

Among "detractors" of the definition of ontologies given by Gruber, we find Guarino ([Guarino, 1998] and more recently [Guarino et al., 2009]) – who, by the way, is also one of the authors of the DOLCE upper (foundational) ontology, which reflects a thorough and deep philosophical setting-out. His criticism focuses on the notion of conceptualization that is specified within an ontology. Indeed, Guarino remarks, the notion of conceptualization that Gruber refers to is the one presented in '80s by Genesereth and Nilsson ([Genesereth and Nilsson, 1987]) which has also inspired much of the work in Knowledge Engineering and Knowledge Representation. Briefly, they consider:

> A body of formally represented knowledge is based on a conceptualization: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly. [Genesereth and Nilsson, 1987]

The form for defining the simplified representation of a conceptualization that they adopt is then that of extensional structures, so that concepts and relations *are* precisely the sets of objects, data that are recorded in the system as instantiating those concepts and relations.

Guarino argues that such a choice implies that any change in the extensional picture described in such a conceptualization produces also a change *of* conceptualization, that which should be really not necessary according to any usual interpretation of the term conceptualization. It indeed deals more with the world of ideas than with specific concrete situations and arrangements of (maybe virtual) objects. It means – we add – that even the "turn-over", over the time, of the instances of a concept causes an unending change of the reference conceptualization. Imagine a knowledge base concerning human resources of a large company, or any other setting that works on flowing, changing information: it is not the best modeling option to have an unstable reference conceptualization.

Guarino therefore proposes a more complex, rigorous account that allows to keep distinct the conceptualization (intended in a fashion closer to common sense) and the actual states of affairs that may occur and change at any time. Basically his idea is to decouple conceptualization and state of affairs by introducing – beside the extensional structure considered by Genesereth, Nilsson, Gruber and many others – an intensional structure to which the conceptualization is anchored and to which any particular state of affairs must conform to be compliant with the conceptualization. In this way many different states of affairs stay within the unique picture described by the intensional structure, which covers an arbitrary number (as many as may occur) of different extensional, actual realizations of the same concepts and relations. The intensional structure indeed provides the interpretation of concept names (and role or property names) as terms of a human language. In particular, it makes room to properly accommodate the dimension of “concept” (ideas, intensions indeed), as in the semiotic triangle, with respect to knowledge bases. In this way, for instance, a set of alternative extensional interpretations for the same concept are allowed precisely because they are declared to be in the intensional interpretation for that concept, each one for a distinct possible world. And there are as many possible extensional interpretations as many possible worlds one can (or wants to) consider for the same conceptualization.

After all, we may say that the collapse of conceptualizations and actual states of affairs was just a lack of theoretical comprehension and explanation, since, in the practice, the distinction usually observed in knowledge bases between a T-box and an A-box basically corresponds to Guarino’s position: the T-box provides the conceptualization, built in form of terminological axioms that define the terms in the specific vocabulary of the knowledge base (the non-logical part of its language), whereas the A-box provides the punctual information that allows to depict specific states of affairs. Then one may choose whether to consider a conceptualization made only of the T-box or necessarily of a T-box together with an A-box.

On the other hand, an ontology, according to Guarino, is the axiomatization provided (possibly in a strictly formal language) in order to restrict the set of intended models fitting the language in which the conceptualization is described. And again one is free to consider the set of axioms depending on the A-box necessary for the specification of the ontology itself, or otherwise. We may just note that current practice uses different names to distinguish different cases. An ontology made only of a T-box is also called *terminology*, since it provides definitions for the terms specific to the ontology; whereas an ontology made also of an A-box may be referred to as a knowledge base *tout court*.

In any case, Guarino’s observations provide us with some further elements to discuss. Of all his discourse about conceptualizations, the most interesting part is the explanation that he provides for the notion of *ontological commitment*[16]. To put it in a nutshell, the ontological commitment, to him, is the property of any formalized ontology that allows to reduce the variety of possible models that satisfy the language given with the ontology itself (as it is introduced with the terms defined in the T-box, plus the special logical formalism adopted), with the aim to approximate as close as possible to the set of the actually *intended models*, i.e. the meaning intended by who has produced the conceptualization. It will be useful to quickly recover some points. As we said a few pages above, an ontology specifies a conceptualization by means of a language which can be more or less formal. The specification of a conceptualization by means of a sufficiently formal language (a logical language) deserves the name of theory. Now, for every logical language there is an infinite set of models in which the terms of the language can be interpreted – and with respect to which is usually considered the truth of assertions (formulas and propositions) in that language. A theory may provide logical definitions for the terms of the language and in particular, with

[16] We will come back on ontological commitment, and will discuss it more diffusely, in the third part of this work, devoted to philosophical issues.

respect to ontologies, for the terms denoting concepts and relations of the underlying conceptualization, by means of terminological axioms. Each of such axioms contributes to make more precise the interpretation of the single term and of the whole ontology as the complex of the theory – thus also reducing the number of models that satisfy the ontology language constrained to that theory. In general, models stay always infinite for any given theory; nevertheless, what is interesting, restrictions on possible interpretations provide a series of conditions to be checked and verified to have some specific state of affairs to be accepted as a valid model for the theory. To be more precise: the presented state of affairs acts as a partial description of an infinite set of models – which all contain that particular state of affairs – which can be all accepted or refused as valid models for the theory based on the conditions expressed by the theory, checked against the "facts" described for that specific state of affairs. The more an axiomatization is strict for a conceptualization, the more definite, precise, verifiable is its ontological commitment – and therefore more useful, rich and informative is the knowledge base that uses it.

Since ontological commitment seems to be a key element in order to get the meaning of a conceptualization it is worth to better examine that which makes a theory (or a language) to commit to an ontology – not intended here as the formal specification of that theory, but as the (acceptance of a theory concerning the) existence of a number of "things" in the world that the conceptualization deals with. It will bring us to observe the relation between meaning and semantics, language and logic. First of all we note that in SW / KR ontologies, even though the names given to concepts and relations may have a meaning in some natural language such that it can be immediately understandable by a human – thus helping the human to recognize what the people who have defined a given conceptualization meant, what they intended to express –, the provision of terminological axioms is the only way available in order to transfer to machines portions of our knowledge about the world[17]. Ontological commitment in information systems, therefore, is to be considered from this point of view, that of derivability of logical consequences (technically: theorems) based on the theory produced to account for a given conceptualization, which is the only part of the meaning which is accessible to computers. We may then try to consider how things work for human knowledge in human minds, that is, how ontological commitment works in natural languages for us human beings. But we keep this point for the third part of this work and for the time being keep on talking about meaning and semantics for machines.

Now, we have that it is the theory, with its terminological axioms, that strengthens and restricts (in the sense of making it more precise) the ontological commitment of a conceptualization. The linguistic part is neutral on this respect, since it can be given an intensional structure as interpretation even if there is no axiomatization, no definition for the terms occurring in its vocabulary. And it is illuminating that ontologies – as the "richest form of vocabulary" – are used as instrument to help in managing large *corpora*, large collections of (any kind of) documents through automated infor-

[17] One might observe that there are other approaches to Artificial Intelligence which do not rely on formal logics or even on declarative instructions at all. But we are talking here of Semantic Web and Knowledge Representation, and the point is not intelligent behaviour, at some level, on the part of artificial agents, but precisely their use of human knowledge.

mation services which behave as if they would have access to the meaning of those documents. Though, it is possible just *via* the "externalization" of the logical meaning, the semantics, out of the natural language of which are "made" the documents about which the ontologies are. Most of the efforts within the Semantic Web Initiative are oriented along the same direction: producing high quality vocabularies with a deep level of detail in defining the conceptualizations that are (or should be) adopted by a variety of information systems. And then make them all sharing knowledge by means of purposedly designed applications that behave in an intelligent fashion, just because they are programmed in such a way to intelligently handle that information. But, after all, is the programmer who has to be intelligent and foresee which (and how) things can be meaningfully shared, with ontologies that help him in understanding the conceptualizations and make the programming phase less strictly coupled with the single particular systems, since they provide very high level languages to access data.

So it is, at the core, the same idea of Artificial Intelligence, revised and corrected in order to tackle the challenge of the open environment (the Web) and with humbler and more fair goals than the ones proposed for the first thirty-fourty years of AI (i.e. reproducing full human natural intelligence).

Within the common notion of meaning we may isolate logical, formal semantics. We remark that meaning in any form of communication depends on the (logical) theory (implicit and informal, as usually for humans, or explicit and formal, as in the case of SW ontologies) that the communicating agents should at least partially share to have communication success, and which expresses their ontological commitment – what they are inclined to accept as being there, exist and, therefore, liable to predication. We do not hold that there is a remaining part of meaning which is out of the reach of logic, since understanding meaning is eminently a matter of reasoning and logic deals with reasoning first of all. Rather, the point is on the formal aspect. That is, the possibility to formally account for the logical presentation of meaning. This is somehow the bottleneck in knowledge representation, since we cannot – it is not feasible – to formally account for everything that enters into the process of understanding meaning. For instance, there is the pragmatic part of meaning that we have no viable way to formally represent even though we may recognize in it a logical content. In our world, to the definition of an object contributes also – perhaps most of all – what one can do with it. On the other hand, in order to explain the use of objects to a machine we really do not use ontologies nor other models for KR, but mere programming. After all it is since the origins of computer science that the meaning (semantics) goes side by side with programming and at the end of processing we find a meaningful output. The semantics of the procedures executed is given beside the syntax – we can see it like labels sticked to data – and the label of the data which the syntactical manipulation of symbols terminates on is the (computed) meaning.

It is curious then to consider within the domains of computer science and knowledge representation the usual distinction between syntax and semantics. Semantics formally provided for computer languages and information systems is that little part of meaning that we are able to make machines deal with. That is, it is (a poorer version of) meaning syntactically manipulated. And this is precisely the semantics that Semantic Web is about. Ontologies then are not to change anything in this picture; they may rather be seen as a useful technique that allows to reduce the efforts to program, but the final behaviour – what should show the intelligence (of the advanced Semantic Web services for instance) – depends on the program that interact with an ontology.

Anyway, since is just the logically and formally defined part of the meaning that on which we can work with ontologies, it is worth to go a little deeper in considering which are the possibilities of ontologies to express ontologically committing theories.

2.5. Description Logics and "reasonable" ontologies

In what follows we consider ontologies from the point of view that we have presented above. That is, as formal theories intended to express the ontological commitment of the language with which a conceptualization is specified. We refer specially to the standard language for Semantic Web ontologies (namely OWL) and its sublanguages that offer not only guarantees about decidability of the theories that they may define, but also an acceptable tractability – that is, the reasoning ends in an acceptable time. This is interesting as far as is considered the possibility for a computer (on which runs a special software program called inferential engine, or *reasoner*) to compute and verify a number of (pretty standard) check tasks on ontologies, which we may sum up as the *agreement* of a knowledge base with respect to the ontological commitment expressed in its ontology. As a consequence, the OWL-Full variant will be mentioned just to highlight what is in it that takes the reasoning out of decidability.

We have already introduced in chapter 1 the family of OWL sub-langauges (dialects or profiles). We now just recall them and provide some more words about the peculiarities of each one before we dip into a more technical observation of their logical expressivity – on which depends the sharpness of the ontological commitment that they allow to formalize. Originally (W3C Recommendation in 2004) there was just OWL in three flavours (Full, DL and Lite) each included in the previous one; five years later OWL2 was approved and other three profiles joined the "team". They provide web "ontologists" with a basic language to use for developping new languages (those which will be the special languages of ontologies) by combining original terms (the entries in the vocabulary of the conceptualization to be specified for any ontology) with the selected set of logical operators available in the OWL dialect chosen. OWL indeed counts a number of class descriptors (or constructors), a set of operators to define properties, a collection of special datatypes, together with a pair of predefined classes – namely `owl:Thing` and `owl:Bottom` corresponding respectively to the top and empty concept – and yet something else less interesting from the logical point of view. We could say that it acts as a *meta-language* with which to build other languages, like SGML with respect to HTML, XML and many others. OWL does that as a logical language: it offers an alphabet of logical operators and a set of primitive predicate symbols (whose "meaning" is encoded in every OWL parser), together with a series of limitations on their use that participate in characterizing the potentialities of each profile. For instance, the possibility to define a property (that is a binary predicate) as a transitive relation on two given sets is available in OWL-DL, but cannot be used together with cardinality constraints, which are a kind of operators used in concept definitions. Now, the selection of one out of OWL-Full, DL or Lite or OWL2-EL, QL or RL implies a different assortment of language-tools (basically: the operators and the limitations on their use) available to design the special purpose language for an ontology.

Apart from OWL-Full – which could be presented as OWL-DL “unleashed” because it uses all the same syntactic potentialities as OWL-DL but with no limits on the kinds of objects to which they can be applied – all the other profiles correspond to and implement some Description Logic (DL), so that we will present our observations on their expressivity looking at the corresponding DL and its typical syntax rather than using OWL(2) syntax.

Description Logics (formerly known as *terminological systems*) are a family of formal languages, sublanguages (fragments) of First Order predicate Logic (FOL), purposedly developed for Knowledge Representation, after the recognition that frames and semantic networks – the models previously most largely adopted for KR – could be given a semantics by means of FOL ([Baader et al., 2003]). DLs guarantee (in most cases) not only decidability of the theories that they allow to define, but also that the reasoning (inferences computing) will stay in a precise class of complexity. Clearly not all DLs are equally expressive nor equally hard (complex to reason on). Quite the contrary, the reason why there is a number of DLs is precisely the trade-off between expressivity and complexity. So that in designing a knowledge base (or an ontology) one chooses the DL best suitable for her purpose. Concerning OWL[18], its first version just allowed to choose between the DLs “embedded” in two of its dialects (OWL-Lite and OWL-DL) which offered a subtle difference in expressivity and complexity: deterministic, exponential in time the former and non-deterministic, exponential in time the latter. That is not impressive performance, but a number of optimizations in available reasoning tools and some limitations in the use of the available operators from those DLs make the two OWL dialects behave quite well in many cases – and always with the guarantee for sound and complete reasoning. The second version (OWL2)[19] instead is entirely built on a very expressive DL – and therefore offering not so brilliant performances –but it is also “split” in three profiles (sublanguages) which are “high-performance optimizations” designed for three special purposes: OWL2-EL that allows for polynomial time reasoning (ideal for very large knowledge bases where expressivity can be traded-off for tractability); OWL2-QL that allows for conjunctive query answering (that is the knowledge base can be queried as a typical database) and whose hardness is logarithmic in space; and finally OWL2-RL that stays in polynomial time complexity and is proposed as a language to bridge ontologies and rule-based systems. The following schema sums up how all the OWL profiles are related based on the relation of inclusion

[18]Cf. the official specifications of the language by the W3C at `http://www.w3.org/TR/owl-ref/`
[19]Cf. the official specifications of the language by the W3C at `http://www.w3.org/TR/owl2-overview/`

among the Description Logics languages that they implement.

$$
\begin{array}{rcl}
\text{OWL-Full} & \supset & \text{OWL-DL} \supset \text{OWL-Lite} \\
\text{OWL2} & \supset & \text{OWL-DL} \supset \text{OWL-Lite} \\
\text{OWL2} & \supset & \text{OWL2-EL} \\
\text{OWL2} & \supset & \text{OWL2-QL} \\
\text{OWL2} & \supset & \text{OWL2-RL}
\end{array}
$$

We will not plunge too deep in Description Logics details, but we aim at least to give the flavour of such languages so as to have matter enough to raise last remarks.

Description logics are a *fragment* of FOL for the fact that they are languages that express formulas with at most two variables – in this sense DLs are also called *syntactic subsets* of FOL. This is the reason why we have also in Semantic Web ontologies, as in the whole field of Knowledge Representation, predicates which are just unary or binary. We have now also the opportunity to fix some points about the "mixed" terminology that we have used so far. In KR, and DLs, a unary predicate is called *concept*, whereas in OWL and RDF it is called *class*. Such a difference might also suggest some reflections about the shift from intensional to extensional level of interpretation, but we will come back on this later on. Binary predicates are called *roles* in DLs and KR, whereas *properties* in OWL – and we have generally called them relations. Finally, *individuals* may be called thus in both domains, even though in KR they are often called also instances, especially when considered with respect to some concept.

Every DL is defined by the special set of concept and role descriptors that it offers. Such descriptors can be seen as abbreviations of First Order formulas, each involving at most two variables, and express each a schema of axioms. The descriptors are used to inductively generate concepts and roles within DL theories. The more complex are the (formulas associated to) descriptors used, the more expressive is the DL – and therefore the theories built on it – but the same is true of the hardness of reasoning on it. All Description Logics can be seen as extensions (exceptionally as reductions) of the same "progenitor" language $\mathcal{ALC}$, that is Attributive Logic with Complements. The descriptors provided by the basic DL $\mathcal{ALC}$ are presented below, by using standard DLs notation:

- A – the atomic concept. A stays for any unary predicate symbol (e.g. `Person`)
- $\top$ – the top concept, containing every other concept in the knowledge base or ontology considered. Every other concept is subsumed by $\top$. In OWL it is called `owl:Thing`
- $\bot$ – the empty concept. No concept description should yield in it. In OWL it is called `owl:Nothing`
- $\neg A$ – the atomic negation, it denotes the complementary concept (e.g. $\neg$ `Person`)
- $C \sqcap D$ – the intersection, it constructs a concept as the intersection between two generic concepts C and D (not necessarily atomic) (e.g. `Person` $\sqcap$ `Animal`)

- $\forall R.C$ – the restriction on possible values (necessarily from the specified concept C) for the role R, that is it defines a concept as that which can be in the relationship R only with the concept C (e.g. $\forall$`hasParent.Person`)
- $\exists R.\top$ – the restriction on role R requiring that there must be some value for it (e.g. $\exists$`hasParent.Person`). It is known as limited existential quantification since it does not allow to further specify which concept should provide the value for R.

Additionally, every DL has also some symbol to express relations between concepts. Typically, the symbols $\equiv$ and $\sqsubset$[20] enter the syntax of any DL, since they are used to provide concept descriptions and possibly definitions – or, generally speaking, terminological axioms:

- $C \sqsubseteq D$ (a general inclusion axiom)
- $C \equiv D$ (a concept defintion if C is just a concept name and D is an expression that uses one of the concept descriptors admitted)

We may note that actually $\mathcal{ALC}$ offers only concept descriptors, no role descriptors. This does not mean that roles are not expressible; they are just not definable, that is $\mathcal{ALC}$ allows only for primitive roles. In order to design a knowledge base with a DL like that, one must establish a *vocabulary* – a set of concepts and role names – and provide logical descriptions for every concept name in the vocabulary. Obviously there will be a minimal set of (atomic) primitive concepts, i.e. not defined concepts, if we have to respect one of the recommendation usually respected with DLs, that is: not to have cyclic definitions. And another crucial recommendation is not to have more than one description (definition) for every concept or role.

For the time being we have no mean to logically define roles. Nevertheless, contrarily to $\mathcal{ALC}$, the DLs implemented by the variety of OWL profiles are quite rich in role descriptors and in additional properties that can be assigned to roles. In particular we recall, for R and S primitive roles:

- $R \subseteq S$ – that is to say that the DL allows for a hierarchy of roles. It is available in all the OWL profiles. E.g.: `hasParent` $\subseteq$ `hasAncestor`
- R^{-} – the inverse role. It is available in all OWL profiles. E.g. `hasAncestor`$^{-}$, it might be something like `hasDescendant`
- $R \circ S$ – which allows for composition of roles. It is a new feature of OWL2 and especially of OWL2-RL. E.g. `hasParent` $\circ$ `hasAncestor`
- $R \circ S \subseteq S$ – that is a generalization of $R \subseteq S$ which accepts also complex role descriptions as described roles. It is available in OWL2
- symmetry – lets declare R a symmetric role. It is in all OWL profiles

[20]Though $\sqsubset$ may be forbidden in order to have *definitorial terminologies*, since general inclusion axioms – like those which can be written by means of $\sqsubset$ are not *stricto sensu* definitions.

- transitivity – lets declare R a transitive role. It is in all OWL profiles
- reflexivity – lets declare R a reflexive role. It is in OWL2
- functionality – lets declare R a functional role (a function), that is it accepts at most one value. It is in all OWL profiles
- inverse functionality – lets declare R an injection, that is nothing can be a value of R for two distinct objects. It is in all OWL profiles.

Most of them are not really role constructors but axioms that can be "attributed" to a role (transitivity, reflexivity, ...). However they are or are not available depending on the particular DL that one deals with. Moreover, the DLs implemented in OWL offers also some more concept constructors. Namely:

- $C \sqcup D$ – union, clearly it allows to define a concept as the union of other two concepts. It is in all OWL profiles
- $\exists R.C$ – is the full existential quantification, it allows to express which concept is admitted to provide the value for R. It is available in all OWL profiles
- $\{a_1, \ldots, a_n\}$ – enumeration or *one-of*, it allows to define a concept based on the individuals that it is made of (for a finite number of individuals). It is in all OWL profiles
- $\geq nR$, $\leq nR$, $= nR$ – cardinality restrictions (respectively minimal, maximal and exact), they are a sort of generalization of existential quantification, so that it is possible to use them to define concepts like `FatherOfAtLeast1`, `FatherOfAtMost3` and `FatherOf2`, respectively for minimal, maximal and exact cardinality. The third case is built *via* the combination of the other two. It is in all OWL profiles in the non-qualified form and in OWL2 also in the qualified form, that is: one may also specify which concept or concept description must provide values for the specified role R
- $R.e$ – exact value restriction or *fills*, it allows to define a concept based on the *quality* of having precisely the specified value e for some role R. It is in all OWL profiles
- $x.R.x$ – self restriction, it allows to define a concept based on the *quality* of having themselves as the value for the specified role R. It is a novelty of OWL2.

Basically, taking the base language $\mathcal{ALC}$ and adding to it the just mentioned additional constructors for concepts and roles (and ready-to-use axioms) we have the DLs implemented in OWL and its variants, and also possibly a number of other DLs.

Now, what is the semantics of all this? After all, we have walked all this trip in Description Logics in order to grasp the final meaning of web ontologies. We know after Guarino that the meaning of an ontology is the ontological commitment that it expresses for the information system that relies on it. This ontological commmitment is that which an agent accessing a web ontology must understand, and clearly the agent can interpret just the strictly logical and formal part of the meaning – that we have called above semantics in order to distinguish it from the broader notion of meaning. Therefore, it is precisely the semantic interpretation of the theories expressed by means of DLs that which comes to be the ontological commitment of a web ontology. Please note that we restrict the discourse to DL compliant ontologies and their semantics – even though also OWL-Full and RDF have one, which by the way is not model theoretical but graph based – for two reasons:

- it is only for DL ontologies, with their "canonical" semantics, that one can recover all the rich experience with reasoners (formerly theorem provers) and optimization techniques that make really profitable – usable by any computer even outside the special application or service for which the ontology is defined – the knowledge described by that ontology;
- and we aim to show all along this work how another interpretation is possible and which may be its interesting aspects with respect precisely to the most considered and most "appreciated" standard set theory, which works fine for DLs. In particular, we think that the interpretation that we will propose may, at least, suggest some ideas for the development of the so-called unifying-logic layer in the picture of the Semantic Web, by proposing a common interpretation framework through the reduction of knowledge bases (in a very general sense) into geometrical objects, less committed to languages (although formalized) and their problems of translation.

Concerning, therefore, the semantics of DLs ontologies, concepts and roles are given a set-theoretic interpretation, so that every concept is interpreted as a set of individuals, and roles are interpreted as sets of pairs of individuals. The domain of interpretation can be chosen arbitrarily and it can be even infinite ([Baader et al., 2003]). The non-finiteness of the domain looks quite natural as far as we deal with something as abstract as concepts. However if the ontology is actually a knowledge base, and therefore there is also an A-box, the domain of interpretation gets somehow restricted since A-boxes provide the knowledge about specific states of affairs and the domain can be reduced so as to be as large as needed to account for all the individuals described in the ontology, which have to be all explicitly named. In any case, let the non-empty set Δ be our domain of interpretation and $\mathcal{I}$ the interpretation function from an ontology to Δ. An atomic concept A is interpreted as $A^{\mathcal{I}} \subseteq \Delta$ while a role R as $R^{\mathcal{I}} \subseteq \Delta \times \Delta$. Let C, D be concepts; R a role between C and D; the semantics of an ontology generated using a DL like $\mathcal{ALC}$ looks like the following:

$$
\begin{aligned}
(\top)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
(\bot)^{\mathcal{I}} &= \emptyset \\
(\neg A)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \backslash A^{\mathcal{I}} \\
(D \sqcap C)^{\mathcal{I}} &= D^{\mathcal{I}} \cap C^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \forall y \, ((x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}})\} \\
(\exists R.\top)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \exists y \, ((x,y) \in R^{\mathcal{I}})\}
\end{aligned}
$$

To say that two concepts C and D are equivalent then is interpreted as

$$C \equiv D \Leftrightarrow C^{\mathcal{I}} = D^{\mathcal{I}}$$

and the weaker general inclusion axioms (defined by $\sqsubset$) are interpreted like

$$C \sqsubseteq D \Leftrightarrow C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$$

The inferences that a reasoner can draw from an ontology like this depend on the axioms given as concept definitions – and where an A-box is available, also on the assertions therein about individuals. In presence of an A-box, assertions concerning generic individuals a, b are interpreted as

$$\begin{aligned}(C(a))^{\mathcal{I}} &= a^{\mathcal{I}} \in C^{\mathcal{I}} \\ (R(a,b))^{\mathcal{I}} &= \left\langle a^{\mathcal{I}}, b^{\mathcal{I}} \right\rangle \in R^{\mathcal{I}}\end{aligned}$$

As regards the interpretation of more complex concept constructors (available in DLs like the ones implemented in OWL profiles), their interpretation is straightforward according to the same technique, that is:

$$\begin{aligned}(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\ (\exists R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \exists y\, ((x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}})\} \\ \{a_1, \ldots, a_n\} &= \{a_1^{\mathcal{I}}, \ldots, a_n^{\mathcal{I}}\} \\ (\geq nR)^{\mathcal{I}} &= \left\{x \in \Delta^{\mathcal{I}} \text{ s.t. } |\{y \mid (x,y) \in R^{\mathcal{I}}\}| \geq n\right\} \\ (\leq nR)^{\mathcal{I}} &= \left\{x \in \Delta^{\mathcal{I}} \text{ s.t. } |\{y \mid (x,y) \in R^{\mathcal{I}}\}| \leq n\right\} \\ (R.e)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \langle x, e^{\mathcal{I}} \rangle \in R^{\mathcal{I}}\} \\ (x.R.x)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \langle x, x \rangle \in R^{\mathcal{I}}\}\end{aligned}$$

And an "exact cardinality restriction" $(= kR)$ is interpreted as the intersection of a minimal and a maximal cardinality restrictions defined for $m = n = k$.

We may note now that constructors such as the *one-of* and the *fills* force the production of a minimal A-box – that is, make the terminology to talk about some individuals. Concerning the *one-of* in particular, one may look at it from another point of view: it is a concept that is defined only by its extension. That is not a "bad" or wrong thing *a priori*, and it is perfectly compliant with set theory, but it somehow breaks the general setting of OWL ontologies (which derive it from RDF semantics). That is, the rigid distinction between intensional and extensional level that is visible in RDF and OWL – we touched that while talking about RDF (1.2.2) – get suspended and the class of objects perfectly coincides with the concept, which is not the standard way for such languages. We can see that looking at the only dialect of OWL which is not a DL, namely OWL-Full. It behaves indeed exactly as RDF, but it has also a richer expressivity thanks to the concept and role constructors that we have just observed – it just can use them without limitations. For instance, it allows to express something like that

$$C \equiv a$$

which is a sort of bomb. Such an axiom, indeed, is licit in RDF and OWL-Full but it makes to collapse the interpretations of a concept (a predicate) and of an individual (a single element of the domain). It also indirectly causes a violation of that which is a typical assumption for any first order language: that the sets of individual constants and of predicate symbols – and also the sets of individual variables and of logical symbols – are all mutually disjoint. Thus, OWL-Full (and RDF too) might bring us at second order. This is why such languages have no "useful" tool to support reasoning. On the other hand, precisely such languages are able to bypass the extensional principle of set theory and assert two concepts (or roles) to be equivalent in spite of the actual assertions about individuals possibly recorded in the A-box. Rather, such an equivalence imposed "from on high" will force the union of the sets of elements (or pair of elements in case of roles) corresponding to the two concepts (or roles). This behaviour is clearly described in the official documentation about the languages RDF and OWL – as far as OWL-Full is concerned, for instance this in the official document that specifies how to interpret RDF:

> The use of the explicit extension mapping also makes it possible for two properties to have exactly the same values, or two classes to contain the same instances, and still be distinct entities. This means that RDFS classes can be considered to be rather more than simple sets; they can be thought of as 'classifications' or 'concepts' which have a robust notion of identity which goes beyond a simple extensional correspondence. This property of the model theory has significant consequences in more expressive languages built on top of RDF, such as OWL, which are capable of expressing identity between properties and classes directly. This 'intensional' nature of classes and properties is sometimes claimed to be a useful property of a descriptive language.[21]

We note at this point that a mechanism like that may be useful for some purposes on the Web, but at the price of making knowledge representation, and therefore information exchange over the Web still more committed to the linguistic paradigm where the names of concepts are necessary and determinant and, on the other hand, actual resources are just occurring instances of no importance.

2.6. Uses of the ontologies

The ability to talk separately and independently about concepts and facts, predicates and individuals is a characteristic of web languages which is not shared with Description Logics. In order to respect such a difference we will now consider the use of ontologies distinguishing two main general settings. The first one is the use of ontologies within single, closed systems, which at most may export information after processing it – but we do not consider what may happen on it when information "leaves" the system. The second one is the use in open contexts, e.g. over the Web, with multiple ontologies which have to communicate, and the focus is precisely on how may ontolo-

[21] From `http://www.w3.org/TR/rdf-mt/`

gies share knowledge, exchange information. That which concerns this second setting, however, will be discussed in the next part of this work (Part II).

For the time being, instead, we will concentrate on intra-system use of ontologies, which is the case where we can observe how DLs potentialities are exploited. By calling this setting "intra-system" we mean the fact that it typically happens within a specific single system. Be it precisely a single software system or a complex framework composed of an interface ontology and one or more databases which actually contain data (and a series of tools needed for integration), that does not matter: we just consider it as a knowledge base and the services that use the knowledge in it are that which we want focus on in the following two subsections. The first one is devoted to the reasoning tasks, that is the checks, inferences and information retrieval operations that can be executed on DL ontologies as far as they are interpreted as seen above, that is basically as first order theories expressed in a language that is a fragment of First Order predicate Logic. The second subsection is devoted to introduce an alternative form of information retrieval that depends on the alternative semantics of ontologies, that is by interpreting them as directed labelled graph. It is the only semantics suitable for RDF (and OWL-Full). We will also see why, even though the specific querying language (SPARQL) developed by W3C is presented as the language for accessing information in the (open) Semantic Web, it may be considered all the same an "intra-system" technology.

2.6.1. Reasoning on ontologies

Knowledge bases defined in some Description Logic are typically coupled with some reasoner, this last being chosen based on its performances with the specific DL adopted in the KB. Conceptually, logical inferences are derivable after the conversion of the theory embedded in the ontology in a plain first order theory, whose axioms are recovered by *expansion* of both T-box and A-box (obviously whenever an A-box is there). For instance, an atomic concept A becomes a predicate such that one may define the class $A = \{x \mid A(x)\}$ with a neat correspondence between concepts (and roles) and such predicates with respect to the set theoretic interpretation previously shown. The expansion is the process by which every axiom involving a defined concept name gets expanded in a more complex first order formula corresponding to the description of that concept. That is, given the concept C, defined as $C \equiv D \sqcap E$, its expansion produces the formula $D(x) \wedge E(x)$. We may also take $\mathbf{T}$, for True, and $\mathbf{F}$, for False, as the expansions of $\top$ and $\bot$ respectively. In the following we will alternate the use of expanded formulas and DLs synthetic descriptors, based on which form will appear the most suitable to deal with in the particular topic that we will face time by time, looking for the best compromise between instructiveness, clearness and simplicity.

There is a set of typical reasoning tasks that most reasoners can perform over a Knowledge Base. They are typically divided into T-box reasoning tasks and A-box reasoning tasks. We now will deal with both kinds of tasks but only introducing what they are intended to check or find, with no in-depth examination of the techniques that are used to compute inferences. T-box reasoning tasks are mainly used during the designing phase of an ontology. They allow to verify whether the ontology is going to be flawed by some "wrong" concept definition. Indeed, an unsatisfiable concept – that is a concept whose defining axiom contains a contradiction – or a concept that yields into contradiction with some other axiom in the theory would make the whole ontology inconsistent. But reasoning algorithms may also be used on the T-box of a knowledge base in order to improve the hierarchy of concepts originally proposed by designers and/or domain experts. Anyway, we may sum up the standard set of reasoning tasks on T-boxes in a short list. Once a T-box is expanded in the theory $\mathcal{T}$, assumed to be consistent, the reasoner can derive four typical kinds of inferences that are particularly interesting to knowledge engineers:

satisfiability of a new concept C, is the check for the existence of at least a model $\mathcal{I}$ for $\mathcal{T} \cup C$ in which $C^{\mathcal{I}}$ is not empty. Or to say it another way, closer to what a reasoner would actually compute, it checks whether $\mathcal{T}, C(x) \nvdash \mathbf{F}$. Should it entail $\mathbf{F}$, either C is contradictory or it is inconsistent with the theory $\mathcal{T}$;

subsumption of a concept C by another concept D, is to check whether $\mathcal{T} \vdash \forall x(C(x) \rightarrow D(x))$, in which case for every model $\mathcal{I}$ of $\mathcal{T}$ it holds that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$;

equivalence of two concepts C and D, is to check whether $\mathcal{T} \vdash \forall x(C(x) \leftrightarrow D(x))$, in which case for every model $\mathcal{I}$ of $\mathcal{T}$ it holds that $C^{\mathcal{I}} = D^{\mathcal{I}}$;

disjointness of two concepts C and D, is to check whether $\mathcal{T}, \exists x(C(x) \wedge D(x)) \vdash \mathbf{F}$, in which case for every model $\mathcal{I}$ of $\mathcal{T}$ it holds that $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$.

Concerning the strategies used to perform these tasks, it will be enough to consider that, according to the specific DL concerned and optimization techniques adopted in the reasoner, when computing inferences every reasoning task is usually reduced to only one basic mechanism perfectly implemented in the reasoner. Typical strategies of this kind are reduction to subsumption and reduction to unsatisfiability. Reduction to subsumption reduces the other tasks by means of the following equivalences

- C is unsatisfiable $\Leftrightarrow$ $C \sqsubseteq \bot$
- $C \equiv D \;\Leftrightarrow\; C \sqsubseteq D$ and $D \sqsubseteq C$
- C disjoint with $D \;\Leftrightarrow\; (C \sqcap D) \sqsubseteq \bot$

whereas reduction to unsatisfiability uses the following

- $C \sqsubseteq D \;\Leftrightarrow\; C \sqcap \neg D$ is unsatisfiable
- $C \equiv D \;\Leftrightarrow\; C \sqcap \neg D$ and $\neg C \sqcap D$ are unsatisfiable
- C disjoint with $D \;\Leftrightarrow\; C \sqcap D$ is unsatisfiable

Reasoning tasks on A-boxes are the principal way of using a DL knowledge base once it is completed. That is, such tasks allow to access information in the KB and explicitate implicit knowledge about the data recorded in a more "instructive" way than with databases. The main difference between reasoning on A-boxes and querying databases lies in two distinctive characteristics of logical inference in a model-theoretic setting with respect to data-models of databases, namely the *open world assumption* and the *unique name assumption*, the former holding in knowledge bases and the latter in databases. Open world assumption is that which prevents a concept for which are not recorded instances in the knowledge base to collapse into the empty concept $\perp$. Indeed, in an open world the absence of explicit information about individuals belonging to that concept does not imply that no individual belongs to it. And a concept collapses on $\perp$ if and only if it is logically unsatisfiable based on its definition. In databases, on the contrary, absence of information is interpreted as negative information, that is a sufficient condition to answer that do not exist instances for that which has been searched. Concerning the unique name assumption, on the other hand, we have that in a KB two individuals with different names may be collapsed on the same individual, if and only if it is logically implied by axioms in the A-box and T-box, whereas in databases every "entity" must be referred to with only one name, so that different names imply different entities. To put it in a nutshell, querying a DL ontology or knowledge base is not the same as querying a database, since it is not sufficient to read values matching a query pattern, it rather requires model checking.

Also reasoning on A-boxes requires the expansion process to be performed so as to get a first order theory in which every defined concept is expanded in the corresponding formula in which appear only atomic concepts – that is the corresponding predicates. Since A-boxes store assertions that typically use concepts defined elsewhere (in some T-box), to expand an A-box requires also to get back to the original T-box involved. Basic reasoning tasks on an expanded A-box $\mathcal{A}$ comprise then

consistency check of the A-box with respect to a given T-box $\mathcal{T}$. It checks whether it exists at least a model $\mathcal{I}$ of $\mathcal{A} \cup \mathcal{T}$, otherwise the A-box is inconsistent with respect to that T-box;

instance check which is the basic inference task on A-boxes. It replies to the question whether an assertion α is entailed in every model $\mathcal{I}$ of the A-box $\mathcal{A}$ – that is it checks whether $\mathcal{A} \vdash \alpha$;

retrieval of all the instances of a concept; it is the task most similar to standard querying on databases. For a given concept $C \in \mathcal{T}$, it retrieves all the individuals x such that $C(x)$ is true in every model $\mathcal{I}$ of the A-box $\mathcal{A}$, by checking whether $\mathcal{A} \vdash C(x)$.

Beside all these reasoning tasks there are still many other less common both on T-boxes and on A-boxes, which allow for instance to retrieve the "least common subsumer of two concepts" or the closest concept of which an individual is instance. Nevertheless, due to the need for compatibility between the DL used to represent knowledge in the KB and the reasoner used to compute inferences from that – besides the difficulties to make actually a reasoner communicate with the KB ...–; and given that the use of DLs is not mandatory for the definition of ontologies to be used within Semantic Web, logical inference drawing is not the only way to retrieve information from within SW ontologies. In fact, knowledge bases are used with reasoners just within specific systems, and even if results of the reasoning may be exported, or given as answers to a user's interrogation, there is no direct interaction from outside to within the system. It is up only to specialists (knowledge engineers) to preform useful queries or other reasoning tasks to be offered to users. In particular, one cannot foresee in such a scenario any comfortable space for free access on the part of a general web agent.

2.6.2. Querying ontologies

The alternative way of accessing and using the knowledge contained in an ontology or a knowledge base in the Semantic Web is to query it by means of the language purposedly conceived and developed by the W3C, i.e. SPARQL (Sparql Protocol and RDF Query Language)[22]. One might remember of the above mentioned profile of OWL2 specially optimized for querying (OWL-QL), but really it offers just some (maybe very powerful) potentialities to integrate ontologies and databases, by allowing for (almost) directly using database queries (e.g. SQL queries) over the ontology, but always within a tight integration of system permanently coupled, which can be considered as a whole and unique system.

Therefore, the only way to consider a more open access to information is by means of SPARQL. It offers both the query language strictly speaking and the protocol to send queries and receive results over the Web. The basic idea at the core of SPARQL is to imitate a typical database query language. Like a query language for relational databases (say SQL for instance) it fires queries which are pattern for data to be matched against the logical schema of the database, so SPARQL produces structures which are RDF graphs with variables to be matched against other RDF graphs, such as can be any RDF data repository.

The syntax of the language recalls very closely that of SQL and also offers a handful of operators to express special conditions for the `where` clauses. As regards other similarities with SQL queries, the `select` function is also there in SPARQL, and there are alternative special functions like `construct` which allows to build a new data graph with the reported data – that is to say that the triples retrieved by the query are generated in the form of a new autonomous RDF graph. Indeed, the basic pattern to be matched is that of the RDF triple, made of subject, predicate and object. Each of these elements can be substituted by a variable to be bound to data according to the `where` instructions. A basic graph pattern (i.e. a triple) matches a subgraph of the RDF data graph whenever RDF terms from that subgraph may be substituted for the variables and the resulting graph is equivalent to the subgraph.

Just to give the idea, consider an RDF data repository containing the triples corresponding to the information represented in the example A-box provided above, that is, in a smoothed presentation format:

[22] See `http://www.w3.org/TR/rdf-sparql-query/`

```
:me   foaf:name   Marco Romano
:me   foaf:mbox   mromano@uniroma3.it
:a    foaf:name   V.Michele Abrusci
:a    foaf:mbox   abrusci@uniroma3.it
:b    foaf:name   Christophe Fouquere
:b    foaf:mbox   cf@lipn.univ-paris13.fr
```

Subject and object of each triple are nodes of the RDF graph and the predicate is the edge (whose label may well be the term itself, i.e. `name` or `mbox`). Now a SPARQL query like

```
SELECT  ?name ?mbox
 WHERE  {?x foaf:name ?name
          ?x foaf:mbox ?mbox}
```

defines a graph pattern made of two triples where the subject must be the same, the predicates respectively `foaf:name` and `foaf:mbox`, and the objects are the values required by the query. The results set for this graph pattern matched against the above data graph is

```
Marco Romano          mromano@uniroma3.it
V.Michele Abrusci     abrusci@uniroma3.it
Christophe Fouquere   cf@lipn.univ-paris13.fr
```

There is still another fundamental component of queries in SPARQL, as well as in SQL, which is the `from` clause. In SQL it determines in which table(s) are to be found values for the requested attributes. In SPARQL it is not necessary to signal tables, since tables do not exist and predicates (the RDF graph-like counterparts of database attributes) are directly recognized for they are explicitly named in every triple in which are involved. Nevertheless, needs all the same to specify where is the data graph that one wants to query and also from where are all the predicate that one is going to exploit in order to express the query – that is, in which ontology or (RDF:)schema they are defined. Such "places" over the Web are easily identified by means of the namespace technique. Therefore, concerning our example, the namespaces to be signalled are `<http://xmlns.com/foaf/0.1/>`, prefixed as `foaf:`, and the (fictitious) `<http://example.org/aboutMe/>` which is assumed to be the base namespace and in the example query is hidden. By the way, we may remark that individual `me` is known, whereas `a` and `b` are not named individuals, but blank nodes since they appear in the RDF graph just as holders of a (`foaf:`)name and (`foaf:`)mailbox.

Given the importance of localizing exactly the namespaces – that which is made possible thanks to their form of URIs (and IRIs), we cannot but note the great implicit limitation of such an approach to data access over the Web. That is once again the need to know in advance the language in which data are recorded. This means especially to know the vocabulary proper to any single ontology or schema from which are recovered predicate names for the triples, and to have an adequate comprehension of the intended meaning – the ontological commitment we could say about DL ontologies – of its designers. In fact, there is no way for the program executing the SPARQL querying protocol to get aware of possible restrictions on the use of some predicate based on the axiom that defines the corresponding concept. Simply because such a kind of reasoning is not contemplated with respect to querying.

Then, one might argue what is the semantics of all that. It is not model theoretic. There is no interpretation of predicate/concepts as sets. It is just a matter of reading the graphs and extracting information (triples) by matching patterns. No hint to understand the meaning (whichever be it) of a predicate or a triple may come from the graph for the computer executing such queries. Rather, for the human supervising the process, the predicate names, labelling the arcs of the graphs, may be helpful. Anyway, yet beyond considerations about the possible reliance of humans on labels in the graph, the point that we see and focus on is the too strict dependence on languages, on the linguistic paradigm. Basically, because it is necessary to know the language of all the ontologies (schemata) involved in order to compose a query. A condition like this makes the use of SPARQL an activity for application or service designers, who write code and possibly embed in it some queries for use on the part of web users, possibly also entering some parameters for the query, but which is fixed, get static as soon as it is written down.

There is little difference then with respect to the uses which are possible within strictly coupled KB-reasoner systems, apart from the specific querying/reasoning tasks that one can perform. The true bottleneck of the whole top-down approach to the enhancement of the Web – we may say the Semantic Web Initiative *tout court* – is that it is a matter for specialists, knowledge engineers or data-repository administrators or any similar professionalism. But in any case advanced services on such a basis of Semantic Web can be foreseen only as purposedly designed programs, written by human people who understand the worlds represented in knowledge bases thanks to their human intelligence, and therefore can set up meaningful services. Thus, Semantic Web looks like a great deal for programmers who can work with new, higher level languages. After all, this is one way to understand Semantic Web. Perhaps it is also the right one, at least with respect to both actually ongoing development and current technical possibilities. But we may also imagine another way to fulfill Semantic Web promises – like the ones in the almost science-fiction article by Berners-Lee. The alternative way that we try to figure out is one where web agents are able to extract some, even minimal, logical meaning out of any sort of knowledge base or ontology that they find in the Web, no matter whether they are formalized in a Description Logic allowing for neat model theoretic semantics or just in a powerful web language producing "only" graph structures.

Anyway, a really dynamic, open and efficient way to meaningfully access data over the Web is still to be invented. We hope to contribute a little to this by suggesting in the next part of this work (Part II) a theoretical interpretation which is, we believe, very simple and still quite meaningful, its semantics being so simple that there is very little need for lingustic determination of concepts. Before that, however, we are going to observe in next chapter the living, most dynamic part of the Web, which is not concerned with technical efforts to achieve Semantic Web, but with the improvement of the users' experience of the Web by means of a social effort, a cognitive and intellectual cooperation. We think that a look on this different branch of evolution of the Web may shed some light on that which still lacks – and therefore should be brought in – to Semantic Web.

3. The World Wide Web social evolution

Ten years after its wide spread all over the World – it was then about the year 2000 – the World Wide Web was already considerably different with respect to its first days, so different that people have begun calling it *Web2.0* as if it was a major new release of a successful software application. Such a difference was not a matter of new technologies and/or protocols – of course development of certain technologies had a role in that, like for instance the language (Ajax) that lets data continue to be exchanged between server and client (website and browser) with no need to constantly reload the whole webpage displayed on screen, so that the user can efficiently interact with the page. But the novelty of Web2.0 has been signed by the new role of users, who have become producers of the Web content. Technology in this case has been perfectly transparent: contrarily to Semantic Web, people working in the Web2.0 do not follow special instructions, are not skilled in any specialized branch of computer science concerning communication protocols or Knowledge Representation.

So, basically, the difference between Web and so-called Web2.0 is about the role of the users, who become also producers of the content of Web sites. In particular we may read this as common people producing resources, data, and also classifying them. Spontaneously. With no special commitment to define a perfect, sound, ontologically correct classification. But anyway providing the Web with a huge amount of data (contact information, addresses, bookmarks, ...) and multimedia resources (from textual blog entries to photos, videos, podcasts) organized according to some classification criterion. Is precisely this spontaneous participation by common people that which lacks in the basic vision of Semantic Web Initiative – which, maybe, was intended to act as a background technical "service" to support common people in better, more easily doing that, but which really stays too attached to technical issues and does not meet the working habits of Web2.0 people.

Let us find in the recent history of the Web the hints and clues that make us hold this position, thus also recovering a more detailed account of Web2.0.

3.1. Social networking and collective intelligence

First of all, Web2.0 does not replace the Web. It is rather an evolutionary branch of the Web – or we could say of the "native species" of the Web, these last being websites, web services, web applications and the like – emerged and grown in continuity and in compresence with the previous Web. The principle of the "survival of the fittest" here has not led into total disappearance of the elder, that which suggests that after all the environment (determined by users' needs) supports both species – although many original (old-)Web services have undergone from marginal to thorough rethinking by learning the lesson of Web2.0. However, continuity and compresence are also the reasons why it is not easy to identify one moment when Web2.0 has begun. Rather, we may consider a period of a few years (around the year 2000) after which the snowball effect generated by some "pilot" experiences which introduced the main novelties of Web2.0 has led to general acquaintance with the existence of this new species.

Key characteristics of the Web2.0, before we try to analyze them more in-depth, can be summed up by referring to the result that it produces: give back to the Web its original nature, that of a networked platform where every node of the net is as important as any other. Where every node has, at least in principle, the same role and possibilities. It is interesting to consider that this was precisely the idea that Berners-Lee had about the World Wide Web in early 1990s according to which every user should have been also an editor of the global hypertext, it was the idea of the "read/write Web" ([Berners-Lee and Fischetti, 1999]). But such an idea has been betrayed by the Web industry, which has produced browsers only to access and read information – to be sold by content providers – not to intervene on that, therefore with no ability to add or modify content. And, we note, now that these potentialities are recovered in Web2.0, they are offered on the server side. Berners-Lee then has even questioned the reasons to talk about a Web2.0 as opposed to his Web, supposed to be the Web1.0. Rather he ascribes the lacks and weaknesses that now Web2.0 corrects to the deviation from the original course caused by business and industrial biases about the Web.

Anyway, in order to recover the main steps that have raised the awareness of Web2.0 (or awareness of *real* Web in honor to Berners-Lee), we propose that which seems to be the first definition of Web2.0.

> Web 2.0 is the business revolution in the computer industry caused by the move to the internet as platform, and an attempt to understand the rules for success on that new platform. Chief among those rules is this: Build applications that harness network effects to get better the more people use them. (This is what I've elsewhere called 'harnessing collective intelligence'.)[1]

It is interesting because it is proposed by Tim O'Reilly[2], who is credited to be also the inventor of the expression Web2.0 together with Dale Dougherty when they ideated the (first) Web2.0 Conference in 2004[3]. So, the name for the phenomenon was invented with some delay after it had grown enough to be recognized. The turning point indeed is located, again by O'Reilly, at the burst of the *dotcom-bubble* in 2001: after that moment Web2.0 sites, services and application were clearly recognizable in the panorama of the Web. A little curious is the fact that this definition appeared only two years later. Indeed, it appeared only at the end of year 2006 in reply to a blogger who challenged

[1] From a blog entry at *O'Reilly Radar*, `http://radar.oreilly.com/archives/2006/12/web_20_compact.html`.

[2] O'Reilly is founder and president of a communication company in the USA – O'Reilly Media – specially focused on technologies, computers, Internet and the Web.

[3] Dougherty is co-founder of O'Reilly Media. Concerning the Web2.0 conferences, see `http://www.web2summit.com/web2010`.

O'Reilly to give a real definition of Web2.0 in order to make cease the debate about what actually that expression was to mean. By the way, precisely this ability to follow, at the distance of about five years, the whole debate is a testimony: it gives something of the flavor of Web2.0, where the definition of such a phenomenon lies not in an official document, in the proceedings of a conference or in a dedicated article but amongst comments of a blog entry[4].

Let's now get into the matter of what is really Web2.0. The definition above talks about a business model. Indeed, the turning point has been set in year 2001, just after the dot-com bubble had burst. The focus is then on how to approach the Web as a unique environment whose nature has to be understood and respected in order to succeed in making business of it. In the autumn 2001, in fact, many companies, which had gone on the Web in search for easy earnings thinking of it as a new Eldorado, went bankrupt and busted up. With the benefit of hindsight, economists, analysts, investors and entrepreneurs realized that there was a number of characteristics, in between technological aspects and social mechanisms, that had protected survivor companies, selecting them as the fittest for the Web. The complex of good characteristics is then assumed to be the right approach to the Web and it requires to consider the Web as a platform – a word somewhat abused we think. However, in order to unveil which are such characteristics we recover the oppositions proposed by O'Reilly that allow to compare good and bad practices on different issues[5].

Web(1.0)	**Web2.0**
DoubleClick	Google AdSense
Ofoto	Flickr
Akamai	BitTorrent
mp3.com	Napster
Britannica Online	Wikipedia
personal websites	blog
evite	upcoming.org and EVDB
domain name speculation	search engine optimization
page views	cost per click
screen scraping	web services
publishing	participation
content management systems	**wikis**
directories (taxonomy)	**tagging (folksonomy)**
stickiness	syndication

We will not discuss all of these oppositions; we will focus on a few (just the ones in boldface) that we consider specially relevant to our purpose of comparing Semantic Web and Web2.0. However, a few words about some other oppositions from this list will help in approaching also our core issues.

By contrasting exemplary cases of websites and/or Web companies – like in the first group of oppositions – O'Reilly makes emerge by difference those which are the winning points of the Web2.0 species. The second group of oppositions tackles directly the

[4]See `http://radar.oreilly.com/archives/2006/11/web-30-maybe-wh.html`
[5]Cf. `http://oreilly.com/pub/a/web2/archive/what-is-web-20.html`

capabilities and behaviors exhibited by Web2.0 services and applications in contrast to elder ones. So, while a personal webpage of the former times was a sort of showcase, where the owner could write everything she thought as in a soliloquy in front of an intangible audience, nowadays web-logs (blogs) still allow everybody to publish everything they want, but also enable a dialectic form of communication so that readers may answer, comment and develop a discussion which sometimes is much more interesting than the original blog entry. Consider for instance the case just mentioned of the discussion from which has emerged the definition of Web2.0.

Flickr, *BitTorrent*, *Napster* are all examples of users providing content to the Net. In fact, it needs a precision here: apart from Flickr, the other two examples concern the Web only partially, since they are devoted to file sharing over Internet and the Web is just one possible place to make users meet and share. Beyond legal issues concerning the protection of the intellectual property of the shared files, that which is interesting in such systems is the way how the network is used. Indeed, they show another sense in which all the node are equally important in the network: being all able to upload and download, give and receive, they set up a decentralized network, with no, at least in principle, central node dominating the net. Flickr instead is more closely related to the Web, since it appears as a website (the other being protocols and programs first of all) where users expose and comment their own photos. That is strictly the case of user generated web-content, for it stays on the Web and enriches the website – we will come back in a while on the complementary issue of tagging the photos.

Wikipedia then is clearly contrasted to the *Enciclopædia Britannica Online* because it, again, presents the case of user generating content for a website, but in this case with the purpose of building the “free encyclopedia”: it therefore touches the field of high quality intellectual content. It is out of our intent for this work to comment on the peculiarities of such a project, but we want to remark the relevance of *Wikipedia* to the idea of collective intelligence ([Pierre, 1994]). The success of *Wikipedia* indeed depends totally on the idea of cooperation, collaboration leading into continuous improvement and refinement. On the other hand we also note that *Wikipedia* can be seen as promoting in the field of general human knowledge the same kind of efforts that support the communities of *open source* software developers, being also the open source world, after all, a form of collective intelligence, yet directed to specific objectives.

Besides its own interest and value, *Wikipedia* is important also because it has promoted all over the World the wiki-model, that is the way of creating a website by letting users to build it page by page, every page being generated by creating a link to it from another already existing page. In its simplicity this idea is very close to the original idea of read/write Web by Berners-Lee, with also, in addition, the possibility to edit others’ interventions and comment on the activity done. O’Reilly contrasts wikis to content management systems. Before we enter more in-depth this opposition, we may locate it within the broader one concerning publishing versus participation, that somehow re-collects many of the other that we have mentioned. It compares the static publication of contents on the part of content providers (usually companies and organization already involved in the cultural industry also outside the Web) and the generation of content (of any kind: texts, encyclopedic entries, photos, videos, reviews, . . .) on the part of common users – which by the way should not be called just users now.

In order to record, keep together and make accessible to other people the content produced in a website – no matter now who has produced it – there are two alternative ways, and then actually a variety of other intermediate ways in the middle. Nevertheless, for sake of the contrastive analysis we are following, after O'Reilly, we look here only at the extremes. On the one hand, then, one may conceive a specific logico-conceptual structure of the website, possibly to be supported by a special software program, the content management system (CMS) which provides a database backend where content is stored and a user interface to output the content requested by users. In this case the logico-conceptual structure of the website depends on its owner[6], who should foresee at his best how people will think when arriving to the website in order to find something amidst all the content stored there. He then will probably offer some search tools, like a basic textual search enegine, which is likely to be already integrated in the CMS. And maybe he will provide a more or less detailed menu of the subjects which are dealt with in the website. In any case, it is the owner who designs all of the structure of the website and people just has to understand and follow. On the other hand, the alternative way is to let people, users decide and define the structure of the website. This is perfectly possible, in particular, by means of wikis, that is the software programs which create an hypertext environment like the one of Wikipedia where every user can add new pages, thus forming the overall structure – therefore the website has no proper structure before people get involved in building it, and it will result directly as product of collective intelligence.

Moreover, in the "pre-formed" websites, for every new content added, who introduces it (owner or external user-collaborator) must also specify where, among the categories available in the CMS it is to be located. In a wiki, on the contrary, the only position to be considered is the location in the hypertext, which is determined by the links incoming in the new page and outgoing from it. If we consider that categories are indicated with a name, a word, a term that should be evocative for the users who read it, the content-placing issue recalls somehow of an opposition between absolute, linguistic and relative, geometric positioning. But exactly the same issue is into play in the phase of content search, where users either scroll a menu and guess-and-try menu items based on their names (which lead to the corresponding categories), or explore the network link by link. Obviously, there is also the possibility, orthogonal to both the approaches here considered, to directly query the website using the internal search engine. That of search engines is not a core issue of this work; we just say that in order to pose a (linguistic) query to the search engine one must already have a somewhat clear idea of what she is looking for and how it can be called by other people.

Even though the wiki is proposed as one of the most representative characteristics of the Web2.0, there is not so a great number of websites that adopt it. This is to say, on the one hand, that some intermediate forms in between wikis and owner directed CMS are the "best-selling" in nowadays Web – where even the (old-)Web species is still alive. But it is also to remember, on the other hand, that the emergence of Web2.0

[6] Obviously in saying owner we may refer to a number of different persons and professionalisms who, case by case, manage that which we are going to discuss, and who maybe are not actually the owner.

has promoted a number of innovations, but one should not expect to find them all, and not all together, in every Web2.0 site / service / application.

Moreover, concerning the way to position and locate content in websites, there is a third way, which is also fully within the wave of Web2.0: that of tagging. It is somewhere in the middle between the absolute postioning in a hierarchy of categories and the geometric, associative linking throughout the web of an hypertext. Tagging is the specific way of giving meaning to resources available in the Web by attaching them free tags. As if we went sticking *Post-it*s onto the objects that we encounter during our everyday activities in the external world, quickly sketching in one, two, three, n words – as many as you want – that which we consider noteworthy about that object. It could be the term that you typically would use to refer to it; or the feeling that it made you to feel when you have found it; or a remarkable characteristic proper to that particular single object that makes it distinguishable among all other similar objects; or, briefly, whatever you would mind about it. Thus is *free* tagging in the Web2.0, as it is practiced in typically Web2.0 sites such as *Flickr* and *Youtube*, and many others. In this case, the use of words, recorded on the tag, is not so committing to a language as it would be for the selection of a category. Besides a number of issues concerning the "normalization" of the written content of a tag – how to deal with multiple worlds, collocations, idioms? and with singular or plural forms? and with typos? just to say a few – the word or words signed on the tag raise also a number of issues concerning their interpretation, since it is not in general possible to know to which part or aspect of the tagged resource the person who has sticked the tag was referring to. It recalls quite closely the argument of the indeterminacy of translation, proposed by the philosopher Quine, with the additional difficulty that there is no community of native speakers (of the language of tags) to be taken as normative reference, that one could learn from. So, we conclude, but just for the moment since next section will be devoted to an in-depth analysis of tags, that the real value of tags is more as marks of use of a resource on the part of an user than as linguistically determined signals.

Yet more interesting is another phenomenon of Web2.0, direct consequence of the practice of free tagging. It is the emergence of the so-called *folksonomies*. A folksonomy – word invented in 2004 by Vander Wal who meant a taxonomy produced by common people (i.e. folks)[7] – is the result of the collection of tags produced by the community of users of any Web2.0 application, like for instance *Delicious*, once it is considered not only as the flat space where tags lye – since they are not in general structured into a hierarchy – but as a complex structure articulated along three axis: resources, tags and users. It is in particular the social dimension of the users that which may provide an ordering among the tags (as the greek radical $\tau\alpha\xi\iota\varsigma$ in taxonomy demands, since it means order). However, we will say much more concerning folksonomies in the following of this chapter, whereas for the time being it is worth to consider the opposite of folksonomies, according to O'Reilly oppositions.

One step back to classical taxonomies then. They are in the Web since first days in disguise of directories. Nowadays those directories play quite a marginal role as search instruments, at least as the "first try". Consider for instance that *Yahoo!* – who most

[7]Cf. `http://vanderwal.net/folksonomy.html`

of all has bet on directories among the largest Web companies – scores constantly (much) less than the reference search engine *Google*. Nevertheless, directories are still there, yet less apparently, and are used as a more specialized tool in case of specialized searches. Moreover, besides the famous *Yahoo!directories* there is also the old Open Directory Project by *DMOZ*, launched by Netscape – by the way, among the ones that have made the Web different from original Berners-Lee's idea with their first widely diffused commercial browser. This project exploits the collaboration of many hundreds of volunteers which examinate the requests for insertion incoming from other users' recommendations or directly from webmasters, so that one could argue that such directories too are in Web2.0 style – even though typical Web2.0 volunteers are somewhat more proactive. Anyway, is not this point that we are interested on. It is the organizing principle that matters: directories are vast taxonomies. They are developed and maintained by human "experts" (volunteers surely do their best) who structure the knowledge of the Web. That is they check the sites, assess them and in case of success add the sites to the directory, positioning them at the most convenient, relevant, consistent place of the directory with respect to the activity, content, or whatsoever that is dealt with at that address. Doing so, they somehow also map the Web. Now, such an activity of structuring our knowledge of the Web is very close to the activity of domain experts, for any other domain, which are involved in ontology design and knowledge base development. Most of all, the basic organizing principle is the same, that of taxonomy, even though in most knowledge bases there is also something more, in particular a definition is provided for (possibly) every category name in ontologies – where in fact one talks about concepts (and definitions of concepts) rather than categories. So one could quickly remark that the core model for representing knowledge in Semantic Web resembles more closely to the (old-)Web paradigm than to Web2.0 as regards its fundamental organizing structure – "just" taxonomy, although Semantic Web and Web2.0 are about twins, having been declared born officially in the same year (2001).

On the other hand, we may readily signal another aspect for which Semantic Web resembles to Web2.0. Indeed, Semantic Web ontologies, thanks to the namespace technique, are able to import from each other concepts and their definitions and therefore to reuse "matter" (formalized knowledge) produced and published by other people. That is also typically Web2.0. One finds it also in the list of oppositions by O'Reilly as the couple stickyness vs syndication. Stickyness stands for the static publishing of contents on a website. The resource is there and who is interested must go there and see. Web2.0 websites instead are often designed in such a way to have content flowing from other websites and possibly going still further to other websites or application. The height of syndication is people that associate to a service or website, subscribe to it and directly receive in their own site or in a desktop application the latest news and updates about the subjects that they are interested in – it happens by means of the well known RSS feeds. The history of the RSS format for delivery of web content is somewhat troubled and it is quite surprising that, in spite of that, it is actually largely used. In fact, there is no one single RSS format, but a quite large family. W3C started to work on it at the end of 1990s and released the first – and only – standard version (a W3C recommendation) in 2000, naming it RSS 1.0, with RSS standing for RDF Site

Summary. The name already signals that such a format basically relies on the RDF language which is a key element for Semantic Web. Thus, in the vision of W3C, the publishing and transmission of web content to be directly interpreted by web applications, could have been a first onset of Semantic Web. Nevertheless, one year before, *Netscape* had already released a version 0.9 of RSS, developed independently. Moreover, few months later published another version (0.91) from which RDF elements had been removed – it is based just on XML – and the acronym RSS was reassigned to Rich Site Summary. Additionally, at the end of year 2000 another company (*UserLand*) especially involved with blogs, released yet another version, called RSS 0.92, and whose acronym was to be read Really Simple Syndication. The resulting situation is awkward because such all these versions are not fully compatible with each other, so that developers who implement syndication by means of RSS usually have to work thrice. Subsequently, an attempt to "reconcile" the various languages for content syndication on the Web has led to yet another language: *Atom*, which is entirely in XML, that is it uses no RDF. The work on Atom started in 2003 and now it is (since 2005) the language endorsed as the standard for syndication by the Internet Engineering Task Force, another standards organization that tightly cooperates with the World Wide Web Consortium. Thus developers, but also bloggers, may choose in principle within a range of four alternative languages to feed their content in the Web, two of which honoured of a standard recommendation by international organisms, though actually the fundamental choice is between RSS and Atom.

For us, the interest in the – here really concentrated – history of RSS, and Atom, is in just one aspect: the independent development of the format on the part of the companies then most active and interested in syndication. They have not waited for W3C to release the standard, rather they made it themselves. Even though this has led to the current chaotic situation, it demonstrates that companies have speeded up the W3C standardization process of RSS – which was one related to Semantic Web – because they were strongly interested in it. On the contrary, we cannot see the same occurring at present concerning other parts of the Semantic Web project, and in particular with respect to ontologies. They are mostly studied and developed in research organizations and do not have direct impact on the users' everyday web experience.

We acknowledge the ongoing diffusion of some ontologies which have been developed spontaneously by communities on the Web – one for all: FOAF – but we lament the lack of corresponding tools and, most of all, clear ideas about how to use them. We note in addition the existence of relatively small movements which act somehow in parallel to W3C, by proposing special standard formats devoted to quite specific applications. It is the case for instance of *microformats*, who claims that its formats (among which *hCard*, *hCalendar*, *hReview* and others) are already used in over two billions webpages (as of december 2010). The following are the few words introducing microformats at their homepage:

> Designed for humans first and machines second, *microformats* are a set of simple, open data formats built upon existing and widely adopted standards.[8]

[8] See `http://microformats.org/`

They say "designed for human first" because the information encoded in such formats stay clearly readable and accessible, and also the production of *hCards*, and the like, is an operation that can be easily performed by anyone with a personal space on the Web. Indeed, microformats are really simple to use: they basically are collections of additional XHTML tags (that is HTML powered with some capabilities of XML), so that to use such formats requires no experience with Semantic Web technologies[9]. As a consequence, these very simple and stable formats are quite quickly spreading over the Web, allowing to make recognizeable – and therefore directly usable by purposedly designed application – certain kinds of data, like for instance personal contact information in *hCards* (that is, an alternative to FOAF). This is the "machines second" part, for which *microformats* seem to propose their formats almost in competition with Semantic Web, so as to demonstrate that it does not need experts and specialists but just common people to make web documents understandable by machines, and so have computer processing and providing users with elaborated data retrieved on the Web. Briefly, it looks as if this people is not willing to wait for Semantic Web to come and have decided to make it themselves, by using simpler technologies, more easily embeddable in current websites and services. We may then hazard one provisional conclusion about the relationship between Semantic Web and Social Web. That is, Web2.0, or Social Web, looks like a restless, more active and vital than Semantic Web Initiative, and Web2.0 enthusiasts are anxious to see the results of every innovation that appears on the horizon, so that in some fields they even try to bypass the work by Semantic Web people and look for their own solutions to problems like, for instance, the reuse of published resources, the delivery of Web content, or the signalling to browsers and other web applications of single pieces of information within webpages – thanks to a well defined semantics, which is directly defined in the code of the application that will use such information (hard-coding which however seems to be inescapabale also for pure Semantic Web solutions). For special issues like these, Web2.0 looks like abusing of Semantic Web technologies with the aim to do something equivalent, maybe even better, on its own, without any further wait to see Semantic Web becoming reality, rather producing it in an alternative way.

3.2. What is a tag?

Since it is a relevant part of Web2.0, the most significant for us as it enables an original form of knowledge representation (i.e. folksonomy, to which is devoted next section), we now concentrate on tags and *free tagging*. First of all we signal the importance of the adjective "free", as it distinguishes from other forms of tagging that have a much longer history and for which does not apply the most of what we are going to observe. Hence, free tagging is the act of annotating a (web-)resource – a photo, a video, a post in a blog, a comment to a post, a webpage, . . .) – by means of a tag, that is, basically a label with something written on it, with the purpose to categorize it within a specific environment – which however could even be the whole Wordl Wide Web. Free tagging is performed in many web(2.0) sites or applications like *Delicious*, *Technorati*

[9] Alternatively they can be expressed in other languages such as for instance RDF.

and *Flickr* to name just some of the best known. It is *free* as far as people may write on tags whatever they want – no need to say that anyway it should be something relevant to the resources to be tagged. Then, the difference is with respect to tags, or labels, as they are used in quite different systems of categorization of documentary resources, like in librarianship where is a professional who decides the right categories for documents, in which cases however one would preferably talk about facets. On the contrary, free tagging is free also because everybody can stick tags on everything. Most of all, while a librarian must respect the specific controlled vocabulary, or special catalogue of category-terms defined some day by a panel of experts, with free tags there is no underlying assumption about what could be written on them.

Now the core point: what is precisely, deeply a tag? The answer we will give is reinforced by an interesting work by Monnin ([Monnin, 2009]) who proposes an idea very close to ours own on many respects. By the way, we believe that our proposal for an alternative logical representation of both ontologies and folksonomies – we will introduce it in next part of the work – is a stimulating contribution to an issue that Monnin leaves open concerning the peculiar relation between ontologies and folksonomies, or more generally free tagging.

The most common reply to the question about what is a tag focuses on the triple relation involving **user**, **tag** and **resource**. However, that is not a defintion of tag, rather of tagging. It indeed introduces the elements that make tagging occur: a user who assigns a tag to a resource. Moreover, such an explanation makes to collapse two totally, ontologically different parts of the tag, that is the access relation (to the resource) on one hand and the referential, symbolic relation to some concept on the other hand. Maybe this is partly due to a rough similarity with the well known semiotic triangle, usually taken as a fundamental reference also in Knowledge Representation, where a **concept** defines a class of **objects** by means of some linguistical formulation, typically the **term** that signals the concept. At the basis of all that there is clearly the triangle mind - language - reality. Given that the involved elements are just three and that tags also deal with classes of resources by means of the written content of a tag – used as search key by the search engine that empowers any tagging environment – it is quite easy to slip into the general semiotic triangle and substitute **concepts** with **users**. After all, users should be the holder of the minds who assigns tags, so that the basic triple mind - language - reality yet holds.

As we said in the previous section, a good metaphor of free tagging seems to be that of *Post-it* notes. Surely it is illuminating to unveil the two distinct relationships that we have mentioned: that of (technical) access and that of (semantic) reference. Indeed, before the written content of the note on a *Post-it*, there is that piece of paper. Equally also the free tag is made of two parts: the “white surface” on which one can write something, which is the tag strictly speaking, and the written part, which we could call label in order to keep them distinct. Now, the access relationship is that peculiar mechanism that allow to reach, to get in contact with a resource by selecting its tag. In the case of a *Post-it* it depends on the glue belt on the rear of the paper, whereas in the case of web tags it depends on some more complex technology that allows to record the label (the written part) as meta-data of the resource. Due to immateriality of web tags it is difficult to talk about them leaving aside the labels; nevertheless, if the labels are recorded as metadata and become the values in special fields among the descriptors of a resource within an application, the tag then is the empty field, the space where the value is placed. As it happens with any Web language, from HTML to OWL, the tag is a signpost, whereas the label is the content for the signpost. This is a trivial example from HTML:

```
<title> I'm a label in a tag <\title>
```

In spite of some slightly uncomfortable shifts in terminology, we may say that the tag is both the whole line and each of the signposts enclosed in angle brackets (that is the opening `title` and the closing `\title`), whereas the free written content is the informative part which should help a user to identify what the tagged resource is about – in this case, it should present the title of a webpage. However, free tags are not supposed to be anything like the title of the resource – but it could even be the case. The point is precisely that it is not defined. The access relationship then is something heavily dependent on the special technology used within an application or website. On the contrary, the label, the written content, is independent from it. It rather brings to the realm of ideas and concepts by passing through a language. It needs no special technology, apart from that of human brain that associates words and meanings according to semantics. Reference relationship then is somewhat independent of the tag. It may occur even outside the tag, without any supporting tag.

We have then a loose relation between label and resource, which is supposed to be informative about the nature of the resource; and a tight relation between tag and resource, which links them materially: the tag is generated in form of a new line in the metadata file associated to the resource (or directly attached to the resource, according to the particular technology adopted). In any case it is some piece of code steadily associated to a resource, which offers a deterministic, material, causal connection between tag and resource.

The question that emerges is then: which one is the most interesting part of a tag? Clearly the label is that which is used to aggregate content – when used as a search key in order to retrieve resources – and also it is the label that which a user wants to communicate to the rest of the community. Nevertheless, the meaning of a label stays extremely uncertain, for at least two orders of reasons. On the one hand there are purely syntactical difficulties, which are dealt with in different tagging environments in a number of different ways. For instance, the number of words that can be used in a label; the way how to write collocations, that is expressions made of more than one word which express a single untary concept; the possibility to have multiple labels on the same tag or to be constrained to "one tag, one label" limitation; the risk of mistyped labels. All these make difficult to find what is the term to be interpreted.

On the other hand there is a major philosophical issue: that of indeterminacy. Being there no assumption on what the tag should be about – apart from being relevant to the resource – one cannot know whether the word(s) in the label refers to the essence of the resource or to any additional, accidental quality; or even to a personal judgement about the resource on the part of some user. We have already touched this point: it is the age-old problem of the indeterminacy of interpretation. It makes difficult to find which part of the tagged resource should help in the process that associate the tag-term, the label, to the meaning intended by the user who has assigned the tag, the concept that she had in her mind concerning that resource. To find which is the right part one has to access the resource and touch it, watch it, read it, or what is possible depending on the nature of the resource. Briefly: one must use the resource and recover from it, find in the resource itself that minimal context that is necessary to reach for a full understanding of the meaning of the label. We strongly remark this need for the use of the resource.

Figure 3.1 – loosely inspired by [Monnin, 2009] though more detailed and "stretched" so as to encompass the conceptual dimension – will help to clarify the discourse. Much better than the semiotic triangle, it illustrates the nature of a tag.

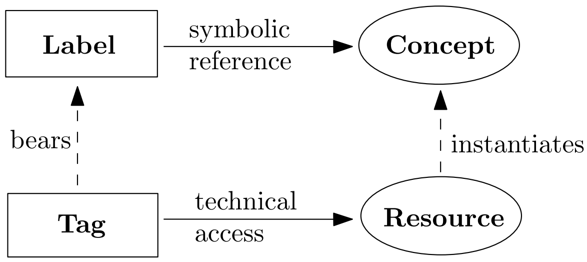


Figure 3.1.: The double nature of tags

From it, the double nature of a tag is apparent. Moreover it allows to see why the reference relation is so loose: it does not hold directly between the label and the resource, but between the label and a reference meaning, a concept that is instantiated in the resource. The reference relation should be then represented as the composition of other two relationships, as the diagram suggests

$$\text{Label} \xrightarrow[\text{means}]{} \text{Concept} \xrightarrow[\text{exemplified in}]{} \text{Resource}$$

which by the way is very close to, once again, the semiotic triangle. It is about the same that accounts for Semantic Web ontologies interpretation, apart from the fact that there is no special selection of tags to be used with tags. Conversely, what are Semantic Web ontologies (as knowledge bases in particular) if not collections of tagged resources? Just, in this case, with a very well defined vocabulary from which tags are to be chosen in order to categorize resources – it is interesting that with ontologies one prefers to say classify instead of categorize: it is sounds of higher quality. Indeed, the vocabulary given together with an ontology provides also logical meaning to concepts, so that even a machine can handle and partly understand them.

Nevertheless, we remark that should not be attached too much importance to the semantics of tags. A tag indeed is yet a tag even if the label that it bears has no meaning; even if the written string is not from any human language, so that it has no natural meaning, no accessible semantics. The idea is easy to be accepted: there are already some kinds of tag that do not bear any natural meaning, that is do not mean anything in no natural language. Such tags are known as *triple tags*, or *machine tags*. Triple tags stresses the way how labels appear on them, whereas machine tags stresses the main use of such tags. Indeed they are used within a few applications, like *Delicious* and *Flickr* for instance, and have a precise meaning for machines rather than for human beings. However, they work exactly as any other tag used in the application: there is a part (the label) where the user writes that which she considers relevant; and the tag can be used to access resources, all the ones that bear the same value for the tag. Indeed it is a value. The label of a triple tag has the typical form `namespace:predicate=value`. While the `value` is the free part of the label – as far as the user can fill in that place with a free string[10] –, `namespace` and `predicate` clearly recalls of Semantic Web technologies and let figure out of a sort of vocabulary defined somewhere (the `namespace` bringing there) and where a set of terms (the `predicates`) are defined, the reason for which an application can understand the meaning of such tags. Just to give the idea, a triple tag looks like `system:filetype=audio`, which actually is a working tag in *Delicious*. To be honest, labels like the one of the example, although being not expressed in a natural language, yet can be easily understood by humans. But the general case is much more interesting: in spite of the attention paid to its semantic content, it is not the most important part. After all, there is no assumption about that which is to be in the label of a tag. Rather, the worth of a tag is in the fact that it acts as a handle to take resources, both on the part of human users and, possibly, on the part of special bots – web agents not so general-purpose, but quite the contrary, specially designed to accomplish some specific tasks.

In the following we will concentrate just on the lower part of the figure 3.1, so as to leave aside all that is concerned with languages and their semantics (be they natural or artificial). Indeed, we think that semantics is better dealt with in ontologies, formal vocabularies and the like, whereas it gets only an approximative, rough management in tagging environments. This is why we aim to better investigate both the access relationship between tag and resource, and also the role of the use of a resource by a user, which – as we have seen just above – is a necessary element in order to really understand the meaning of the label on a tag, since it provides some form of context to the communicative act recorded in a tag. Indeed, the use of a tag is a form of distance interaction between a user who assigns the tag and any other user who meets the tagged resource. Hence, for the communication to succeed one needs to take into consideration all the context, in which are surely to counted both users besides the tag and the resource – and for sake of exhaustivity one should consider also all the other tags on that resource (assigned by the first user) and the other resources bearing that tag (again, assigned by the same user). So, it seems that the Web2.0 calls for something more than mere (formal) semantics to get adequtely understood and explained. It needs something like a pragmatics of the Web, a study of the relations of use that emerges and gets reinforced in social environments. After all, many of the formal studies conducted on folksonomies (we will see them more closer in next part), although

[10] Actually the value string could be not so free to choose. Precisely for the coded semantics of such tags, may there be a closed, predefined set of possible values, as is the case with *Delicious* file type tags, which simply say what kind of digital resource is the one tagged, out of the list of file extension types supported (e.g. .pdf, .doc, .mp3).

aimed to extract traditional structures out of folksonomies like ontologies or at least hierarchical taxonomies, focus precisely on the social dynamics that take place in a tagging environment. However, they focus mainly – if not exclusively – on the use of tags on the part of users, since usually people stay interested in finding the semantics emerging from social environments. On the contrary, we will try to focus on the use of resources, searching for insights about classifications of resources, objects, data, based on a general notion of compatibility. Anyway, this will be the point of the second part of this work.

We conclude this short analysis of what is a tag – as a single, complex artifact – with another consideration about the relation between Web2.0 and Semantic Web. Indeed, here again, in tagging environments, we find a situation in which Web2.0 puts in function something that has the flavour of Semantic Web, anticipating on W3C, but which really is a direct answer to an emerging need – or an enriching proposal to common users behaviour – that arises from everyday use of very conceptually simple instruments, although it can be realized by using W3C standards and Semantic Web technologies.

3.3. Folksonomies

Let us now consider what tags, when they are a vast collection, enable. That is, we are going to see more closely the new approach to knowledge organization known as *folksonomy*, which apparently has emerged in Web2.0 tagging environments.

First of all we have to place it in the context of the other methods and approaches used, in the Web, for knowledge organization. Really they are not so many, alternatives being just two – besides folksonomies themselves. The main, most largely used and best known is the hierarchical, taxonomical system. It is the one usually "integrated" also in ontologies, although an ontology could be designed even in non-hierarchical way. It produces, in the simplest case a typical structure recognizeable as a tree, with a root, representing the whole domain of interest, and a number of branchings that progressively specialize the matter. To each node correspond more or less roughly a concept of the domain; concepts on the same branch are related to each other according to the is-a relationship (or parent-child), which precisely represents the specialization process, from the root to the leaves. At the opposite end from the root, there are leaves, that is the most specific concepts. In case the taxonomy is a tree, there is only one path from the root to any leaf. But really ontologies may draw more complex graphs, so that even cycles may occur. In any case, as regards the hierarchical-taxonomical model for knowledge organization, once the structure is given, the items (web resources for instance) can be placed in one and only one node, preferably into a leaf. The tree-like structure indeed promises a rich payoff for the careful placement of items: the "container" node – typically identified by a word, but not necessarily – in which an item is placed provides the item with a clearly defined "meaning", encoded in the structure of the taxonomy itself. What may be the meaning of an item classified in this way depends on the domain to which is committed the taxonomy and on special choices taken by the professionals who have produced it. For instance, hierarchical taxonomies are used to classify books in librarianship, so that the specialization criterion may be that of the subject matter of every book-item. Given that in hierarchical taxonomies an item occupies only one position in the tree, it is apparent that not only the designing of the taxonomy, but also its continuous use for classification of resources requires a professional or at least somebody who is trained in doing that, somebody who knows quite well the represented domain. Ontologies on this respect are much more flexible since they allow for multiple classifications – nevertheless, they need professionals or experts because of the accuracy required by the presence of logical axioms to be respected.

Yet being a highly effective approach for knowledge organization, hierarchical taxonomies have some important limitations (cf. [Quintarelli, 2005]), among which we highlight:

- taxonomies are static, cannot be updated to better account for any novelty. Once the taxonomy is designed, it cannot be modified unless resetting the whole taxonomy and re-classifying every item;
- strictness of taxonomies may cause imprecision. It is the case for an item which could be classified under more than one concept / class / category;
- taxonomies are close-minded, since they are built on the cataloguers' point of view, which use their way of thinking, knowledge about the World and words to express it;
- lastly, taxonomies require homogeneous resources to be classified; they are not suitable for heterogeneous resources.

Once again, we care to remark that ontologies usually follow the hierarchical taxonomical way of arranging the concepts of the domain that they are to describe, but they are not necessarily, strictly speaking hierarchical taxonomies, so that none of these shortcomings necessarily holds for an ontology. Nevertheless, hierarchical taxonomies are still widely adopted, more or less consciously in the Web, not only in digitalized libraries. Consider for instance the number of webmasters (or their software counterparts, CMSs) that in many websites may offer nothing but a list of maybe nested categories as the only way to classify items, e.g. blog entries.

The other approach to knowledge organization for digital resources, which sometimes "appears" on the Web but is more often used within special, closed applications, is that of facets. We can sum up it as being the same as tagging, but with a predefined set of possible labels. In faceted-systems, indeed, classification of resources is performed by labelling them with keywords, that is basically the same as sticking tags bearing labels on them. But it is not free tagging: the set of allowed keywords is defined by experts who select relevant keywords, like the same names that could be used in a taxonomy. The main difference with taxonomies is in the multidimensionality: a taxonomy is articulated along only one criterion that leads from the root to the leaves by specializing the categories, and the same notion of specialization requires that categories are homogeneous, not only items, so that it makes sense to have a category $C1$ to be specialized in two categories $C2$ and $C3$. With facets, on the contrary, there is no requirement for homogeneity of categories, so that keywords may be articulated in an arbitrary number of dimensions. Sticking to the general case of librarianship, in a faceted-system resources can be classified by type of publication (book, article, ...) and by subject matter orthogonally. In this case, every item can have many facets (not only one category), up to one for every dimension that is contemplated in the system. Also, every label / keyword can be used as a search key for retrieving all the resources within the system that are marked with that label, and multiple keywords on resources allow to interrogate the system by means of complex queries in which multiple keywords are combined. Moreover, such systems can be updated since it is possible and quite easy to add new keywords, or even entire new dimensions – although resources entered and classified before the update will have no value for that new dimension.

Nevertheless, also faceted-systems have their shortcomings. In particular, they too, as well as hierarchical-taxonomies,

- are "close-minded", since the fixed set of keywords is defined by a pool of experts professionals;
- require skilled people to operate the classification of resources;
- and require homogeneous collections of items to be classified.

The last point in particular, depends on the fact that the significance of the facets (and keywords) is related to the set of dimensions (classification criteria) considered in the system, so that they are meaningful as far as the resources which get classified "have" those dimensions.

To sum up, taxonomies provide a hierarchy of categories along only one dimension to classify a collection of homogeneous resources. So, they are rigid both in the structure and in the possibility of use, but offer at the same time the most simple and effective strategy to classify – it needs just to understand the point of view adopted within the system. For this reason they require specialists both for developing the system and for using it to classify resources. Somewhat more flexible, faceted-systems provide a flat space articulated in many dimension, but it stays flat for it provides no hierarchy. Such systems are more flexible as regards the structure, but require all the same professionals both for the preparation and for the use of the system, which indeed is somewhat more flexible, but also somewhat more complex to understand: it needs to recognize the multiple dimensions – instead of only one – along which resources may be classified. Because of the multiple dimensions, faceted-systems can be extremely accurate in classifying resources, which however must come only from collections of homogeneous items.

Continuing the analysis according to these criteria, we find that a folksonomy is the most flexible system: can be used whatever string as the label for the tag, and whatever resource can be tagged. That is true at least in principle, since actually special environments may constrain on the kind of resources admitted. Consider for instance the case of *Flickr*, which is an environment and a social community devoted to photo sharing, as opposed to *Delicious* which deals with whatever in the Web can be referred to by means of a URL.

Let us report an authoritative definition of folksonomy.

> Folksonomy is the result of personal free tagging of information and objects (anything with a URL) for one's own retrieval. The tagging is done in a social environment (usually shared and open to others). Folksonomy is created from the act of tagging by the person consuming the information. [11]

It is the definition by the same inventor of the term, Vander Wal, who coined the term in 2004 (cf. [Vander-Wal, 2004]). Now, concerning the folksonomy as a method, or approach for knowledge organization, it offers something definitely new with respect to taxonomies and faceted-systems. Indeed, besides the multiple dimensions that any user may consider to describe a resource – which may refer to the nature of the resource, to its content, to its author, . . . we have already considered the variety of possibly relevant aspects – there is an additional dimension of the system itself, that is multiplicity of users, which means multiplicity of points of view. Therefore, the problem of indeterminacy of the reference, which affects the labels written on tags making difficult to semantically interpret them, is compensated by the new world that is presented by the social dimension, which enables to consider and observe the use of the resources on the part of users – which is the point that we indicated in previous section. The need for some context to understand the meaning of (labels on) tags can be satisfied in the social dimension. Moreover, the greater flexibility of folksonomies allows a user to use the same tag for disparate kind of resources, provided that they are all in the Web, like for instance a book (through the webpage presenting it), a person (through her personal webpage), a vegetable dish (through its online recipe), and so on. On the one hand, this can be seen as an amplification of the indeterminacy problem, or its persistence at the other end of the reference relation. After all, the problem of indeterminacy is connaturated to the Web, as it is a virtual world talking about the external World, and already for this reason alone it may raise a number of deep, awkward, really ontological questions. But, precisely for that, on the other hand it enables some unforeseen possibilities for ontology discovering in the Web which, we think, have not yet been adequately understood. We can see here into play something like social construction of meaning through social categorization of reality. Or at least an effort in search for an agreement on assumptions about reality, that takes place between any two users that share some tag – where such a sharing appears already in the act of using a tag put by others to retrieve something. It is a sort of bet on the possible meaning of the tag. Anyway, we will deal more in-depth with this aspect in the third part of this work.

For the time being we rather keep on following Vander Wal in his analysis of folksonomies, adding our own considerations. He distingushes two kinds of folksonomies: the broad and the narrow ones ([Vander-Wal, 2005]). The difference lies obviously also in the quantity of resources and users tagging them, but not only in that. Other aspects, that altogether could be indicated as the intention behind the act of tagging, are determinant.

- A broad folksonomy results from a tagging environment where every user tags every kind of resource. Moreover, resources are found independently by every user who tags them. For instance, a user u_1 finds an interesting resource r_1

[11] See `http://vanderwal.net/folksonomy.html`

during his surfing on the Web and decides to save it with a bunch of tags (t_1, t_2 and t_3) in *Delicious*. Another user u_2 stumbles upon the same resource on the Web, maybe having found it as a search result in a common search engine, and considers it worth to be added to her personal bookmarks in *Delicious*. The tagging of that resource on the part of u_2 stays free, even though the system recognizes that the resource is already known in its database (thanks to the tagging by u_1) and may suggest her some of the tags already assigned by others (i.e. just u_1 in this case) to the same resource. The reuse of same tags on a given resource on the part of different users is the aspect of broad folksonomies that is usually most considered. It allows to assess the "popularity" of tags – sometimes observed in combination with the temporal dimension – and then to identify the best working keywords based on the assumption that most commonly used labels refer to most interesting concepts. Or at least concepts interesting for the sub-community of users that intervenes on that resource by assigning some tag. Such dynamics concerning tags and users are also the focus of all current forms of study on folksonomies, that typically aim to extract a vocabulary (and the corresponding conceptualization) by isolating most popular tags (better: labels).

- A narrow folksonomy, on the other hand, results from a relatively smaller tagging environment where users introduce resources (typically contents generated by themselves) and assign tags to it in order to make them findable by other users. In general, users can always add tags also to resources introduced by other people, but if they do, they always stick tags on resources found within the systems. That is, they find the resource in the tagging environment and, in case they think that there is yet some important aspect of the resource that has not put in evidence with a tag, they add yet other tags according to their thought. In this way, the increasing collection of tags on a same resource contributes to an ever more rich, possibly also precise, description of that resource. On the other hand, the social dynamics visible in broad folksonomies get here shadowed: it is not the popularity of tags the major interest – granted that the issue of popularity is interesting to the researchers' eyes, not to the single user who simply aims to organize her bookmarks. Nevertheless, the interesting aspect in a narrow folksonomy is rather the accuracy of tags in retrieving relevant resources. It is a "quality" of the (labels on) tags that every user may contribute to improve by assigning the tags that she considers most relevant to as many resources as possible.

We think that the difference between these two kinds of folksonomies can be understood based on two tightly related factors:

- the presence of (more or less explicit) constraints on the kind of resources with which the system deals – like for the case of *Flickr*, which is specialized in photo sharing, and proposes a narrow folksonomy –;
- the status of the tag itself, the point being whether the tag is considered as an attribute of the resource (like in *Flickr*) or it is considered as a handle in the hands of users (like in *Delicious*).

Of these two factors, the second is clearly the most interesting, but the first is the key element in determining whether tags will be used as attribute of resources or as handles to access resources in the universe of reference – e.g. the World Wide Web for *Delicious*.

In the first case (tags are attributes of resources), the system is somewhat closer to a faceted-system, where a label is an attribute of the recorded item and tends to be about the strictly inherent characteristics of the tagged object, or about its content as a medium for other cultural contents. Therefore, the intention guiding the act of tagging is to provide the best fitting labels for any single resource with respect to its objective reality, its nature, its essential qualities – apart from the inescapable issue of the point-of-view-bias.

In the second case (tags are handles to grab elements in the universe of reference), the system produces that pretty original cognitive environment that we mentioned above and which, we think, would deserve a deeper and further attention precisely as a cognitive artifact. Indeed, in a broad folksonomy tags are not limited to express characteristics inherent in the resource, but may express a variety of different aspects even with only very weak connection with respect to the very "essence" of the resource. For instance, a (label on a) tag could express the particular interest that a user was pursuing during the web-search or surfing when he has found that resource, and for which he has deemed that resource to be worth of tagging, so as to have it at hand next time. Nonetheless, we note that also within a broad folksonomy one may recognize something very close to a classification by facets. It is possible by looking at any portion of folksonomy that consists of all the tags used, and resources tagged, by a single user – that which is known as *personomy*. As its name suggests, a taxonomy made by a single person by means of tags (and their labels) immediately collapses on the model of a faceted-system: just keywords, perfectly managed by a single "expert" – as far as everybody could be considered expert in using his own words and concepts –, the only difference lying in the quality of the single labels used and of the overall collection of labels.

Be it broad or narrow, in any case a folksonomy appears as a much more flexible model and approach to knowledge organization than both hierarchical taxonomies and faceted-systems, mostly because of their "democracy". Everybody, indeed, can assign tags to whatever resource is (or is acceptable) in the specific tagging environment that one considers, without being a librarian or any other kind of professional of knowledge organization or a specialist of the domain to which the tagging environment is committed. Also, whatever string can be used as a label in a tag, even alphanumerical strings with no meaning in any natural language. And possibly not even in any artificial language, as it would be the case for machine-tags. And, finally, every resource (in broad folksonomies at least) can be tagged. This means also that tags may appear in the system together with resources, that is at the same time. Reasonably, a tag could not have a reason to exist in a folksonomy until some resource needs it – or to be more precise, until some user deems that tag necessary for a given resource. Conversely, then, no (label on a) tag exists in a folksonomy unless it is used for at least one resource – that is, there is no "empty category" in a folksonomy. The tight coupling tag-resource depends on the special nature of the objects that are accounted for in a folksonomy: the resources that get tagged indeed are not objects from the material world of human beings, but webpages or other files accessible through the Web, that is symbolic objects – in some sense linguistic they too – exactly like the tags that are assigned to them. Whereas an ontology is usually aimed to talk about the material external world, even though, after all, it has only to feed a computer system with data, a folksonomy avoids the whole set of problems of dealing with external material reality – and most of all of how to get "out there" from within the Web – since it only talks about digital objects, accessible within the Web itself. If one then really wants to consider the relationship between a personal webpage tagged on *Delicious* and the human person about which that page talks, it is all another story and it is not required to investigate such a relation for the folksonomy to be a good folksonomy. Ontologies on the contrary suffer much more of this problem since an ontology cannot escape it in the same way as folksonomies do. The quality of a Semantic Web ontology depends mostly on the closeness of the world description provided in it with respect to the actual world "out there".

All these peculiarities give folksonomies their great flexibility. But, one may argue, it also produce some drawback, especially in using a folksonomy to recover useful information. In fact, the accuracy of a query posed to the system by using any (label of a) tag as keyword will retrieve a set of resources of which one cannot say neither that it is complete – may be there other resources for which that tag could fit, but nobody has assigned them it – nor sound, because users could even "mistag" a resource. Yet, it is hard to define what could be mistagging, since there is no rule to obey in tagging. Anyway we may give that a tag is "wrong" when most of users would absolutely not use it for a given resource. So, it may happen that spammers and / or promoters of some particular product, party or whatelse, mistag resources. Nevertheless, precisely the social dynamics of folksonomies plays as an antidote to such poisoning of the system, since "malicious", wrong tags are outnumbered by the large agreement of many other concordant tags and concordant users – so that, a poet would say, the noise of the few is hushed up by the harmony of many.

Should it be not enough, however, explicit, repressive countermeasures may be taken, as the largest Web2.0 project testifies. Wikipedia indeed started at least in 2006 to systematically protect sensible contents by allowing only selected users to edit most sensible pages, like for instance pages subject to "vandalism", and by blocking the account of users proven to be "vandals" or spammers. And today Wikipedia offers a richly articulated policy for protecting content from ill-intentioned editing. The morals may be that, even though it all (in Social Web) has started with the leading idea of purely paritary communities, the social community itself at some moment – probably when it reaches a sort of critical mass (to be weighted on the number of members) – feels the need to build some protection systems to preserve the wellness of its environment, for instance by introducing a "judiciary system" that judges suspect users' behaviour and punishes when needed. So, we may easily consider that also folksonomies might evolve in a pretty similar way, with "power users" in charge of protecting the environment, maybe by removing bad tags and blocking the users committed to deliberately mistag resources. If this can really be a scenery for the evolution of social tagging communities, then we may also add that broad folksonomies seem to be less affected by damaging activities, almost immune, for they are originally produced, by construction, as collections of personomies, where after all personal and private interest is the added value to the tagging activity.

Clearly, this argument makes sense when the folksonomy is used in the "right" way. Here again, as for wrong above, it is not easy to decide what is right. We may propose as right any form of navigation, exploration of the folksonomy. After all, only this behaviour allows to discover and appreciate the richness of a tagging environment. To query the folksonomy by means of a tag-label used as search key is only the starting point. It is just after that a folksonomy shows its worth from a cognitive point of view, as it allows to pass through a multitude of tag assignements, that is unique relations among resources, tags and users, each one expressing a point of view, each one deriving from a store of personal knowledge and experiences. Hence, the folksonomy allows to touch personal conceptualizations, or even world-visions, underlying any single tag assignement and, briefly, to exchange knowledge with a number of persons.

We can see then in which sense a folksonomy is able to reduce noise, as we stated above: during the exploration of the folksonomy every user is able to see the list of tags for any resource, and a tag-label that clashes with all the other tags used on a resource is very unlikely to appeal a user's attention – unless she looks precisely for marginal, minoritary sentiment, but in that case it is a conscious choice, not a "swindle". Not to say that, being a tag related also to the user who puts it, other users will learn soon to stay away from spammers, or simply will not rely on tags assigned by a user that they do not agree with – exactly like people in real world tend to stay away from people with very different world-visions.

Therefore, we may conclude that folksonomy is a flexible and robust model to organize knowledge, provided that it is used – also in the searching phase – according to its nature, that of a social environment where people share knowledge on the basis of what each one knows. So that, besides finding that which one would expect in response to her queries, she may also discover new points of view on the same things, other respects for which the same resources can be considered.

3.4. Bridging Semantic Web and Web2.0

We have highlighted the value and interest of folksonomies, their originality and the novelty brought to the area of knowledge organization. Nevertheless, folksonomy is not the killer application that will replace any other form of knowledge organization. It is interesting then to consider which relation holds, or may hold, between folksonomy and the other forms of knowledge organization in the Web, that is especially ontology.

It is important to notice (we recall it) that though hierarchies usually do emerge from ontologies, ontology is not necessarily embedding the tree structure that we have found typical of hierarchical-taxonomical approach to knowledge organization. Indeed, ontologies may produce graph structures, with multiple parent-concepts for a given concept. Or even a flat ontology is conceivable in principle – one could then argue for its usefulness. Really, that which matters in (Semantic Web) ontologies is the logical definition of concepts, from which hierarchy emerges, and which by the way provides that part of the meaning – purely logical – that can be "understood" and handled by a machine, a computer program.

On the contrary, folksonomies are not generally considered as bearing any logical meaning. An expression like "the semantics of folksonomy" is something quite puzzling just to hear. Rather, researchers consider the semantics of folksonomy tags, that opens to the area of investigation on how to extract the shared conceptualization underlying the tagging environment. That is identifying the concepts most widely used to make the tag-labels refer to, and then trying to organize them in a taxonomy. The hierarchy that may appear in the resulting extracted taxonomy strictly depends on the use of tags on the part of users, e.g. by observing co-occurrences of tags, that which could suggest the presence of synonyms, hyperonyms, hyponyms – and therefore also super- and sub-concepts. It is then an effort to map, or at least reflect the social and pragmatic dimension – since in folksonomies one can observe just the use of resources and tags by users – into a semantic structure which deals with pure concepts. The main reason that we can see for that is that we have good instruments – both technical and theoretical – and experience to deal with semantics, provided that is intended as the syntacticalized version of the meaning, whereas we have no instruments good the same to deal with pragmatics and sociality. Obviously, this discourse is scoped to the horizon of computer science (especially in the area of Knowledge Representation and information management systems) and of the dynamics that it has to support and possibly improve in the special environment of the World Wide Web, eventually enabling computers to use in a principled fashion human knowledge diffused in the Web.

On the other hand, ontologies totally lack the social dimension. As a consequence, it is obviously a desirable process to bring ontology and folksonomy side by side and let them enrich each other by their peculiar capabilities. It is just a matter of finding the best way to do that.

We have already spoken about the common attitude to recover (possibly) ontologies, or at least vocabularies out of folksonomies. We may consider then an alternative way to bring closer folksonomies and ontologies which has already found its place in the relevant literature. The basic idea is to define an ontology of tagging, although it may assume slightly different forms ([Gruber, 2007], [Sharif, 2009]). In any case it is not the ontology of tags, which would imply to eliminate the genuinity of tagging by providing a limited set of tags, each one corresponding to a concept in the ontology. It is rather the ontology of the tagging activity, able to account for the act of tagging, focusing on all the three elements involved in the activity: user, resource and tag. It also would allow to deal with tag assignments as objects about which further information can be specified. For instance, one may take into consideration the temporal dimension and account for the time when an assignment is produced. Or, as Gruber proposes, one could distinguish between, and systematically account for, polarity of tagging, recordind a tag as positive or negative. That is, once the negation has given a well defined semantic in the system, one may negatively use a tag so as to express that the label used in it does not fit the resource. That which would later enable to query the system about resources which are "complementary" to some keyword, that is bear it negated.

The interest of such and similar additions to the "semantic of folksonomies" lies in the possibility to give them more logical structure, more matter on which a machine may compute something. But also, we note, there is always the interest in determining, in a stable, fixed way, the meaning of labels used in the tag. As it is for instance with another utility that such a tag-ontology should provide: the ability to declare different strings (labels written on tags) as being equivalent, that is expressing the same concept. It is apparent that such a move is intended to recover the ability to deal with semantics of tags in the same way as in taxonomies, besides the help that it may povide in normalizing syntactical variability of tag writing. But in this way, once given a more defined and fixed semantics, tags are estranged from their original context, the only thing that should really determine their meaning. It may look like a betrayal of the very nature of folksonomies.

To sum up, in this chapter, besides presenting the social evolution of the Web that walks along with the technical enhancement directed by the W3C, we have also had the occasion to highlight some cases of "cross-fertilization" between these two. It was the case of the technology supporting RSS, which comes directly from Semantic Web research and enable one of the distinguishing features of the Web2.0. It was also the case of machine tags, which again use Semantic Web languages and / or techniques, such as that of namespaces, to enable special designed services to use single portions of information. We have then remarked, however, that Web2.0 somehow revises Semantic Web according to its own needs and, in doing that, may adopt much simpler architectures – as it is the case with microformats, which are produced even with simple extensions of HTML – that make Semantic Web to look like an oversized ambition.

On the other hand, also Semantic Web tries to take advantage of Web2.0: we have just seen an attempt in this direction with the tag-ontology. Nevertheless, such efforts are marked by the bias of a reduction: folksonomies lose their dynamics, and the social aspect gets shadowed, when the tags are re-organized based on the semantic interpretation of the labels that they carry. A better attempt, we think – and we work for that – would be to develop an approach that provides logical meaning to tag assignments, and then to folksonomies, and still preserves the dynamical, never fixed meaning of tags, always subject to variation being dependent on social dynamics as well as on other contingent situations. Like for instance the extension of the sets of resources covered by the same tag-label, considering that in the general case the wider the set the weaker is the connection, for any single tagged resource, of the label with its symbolic referent (the underlying concept) – but also the more users tag with it, the more its meaning gets reinforced.

In any case, there is no doubt that some form of cooperation between Semantic Web and Web2.0 is desirable and actually needed for further development of the Web. It is enough to consider that the bottleneck of Semantic Web is the (relative) scarcity of annotated data and information, whereas one of the most apparent benefit of Web2.0 is the abundance – really huge quantities – of annotated data. Well, one may claim that is not the same level of quality in annotations – although such a comment reflects the, common and absolutely majoritary, position that "good" annotations are the ones with a neat, well defined standard semantic as it is defined in ontologies. After all, we cannot and do not want to deny that, because that semantics is as good as we have appropriate tools and theories to make good use of it. And at present we have not so good instruments to have machines profitably using folksonomies for instance.

It is for this reason that we aim, with this work, to offer a minimal contribution to find another form of collaboration between Semantic Web and Web2.0, in particular between ontologies and folksonomies. Indeed, in the next part of this work we will lay the foundations for another way to bring ontologies and folksonomies closer. We think in particular that the core value of such a proposal is the fact that it does not imply reduction of one to the other – as on the contrary ontology extraction from folksonomies and also, though less apparently, the approach suggested by Gruber does. Quite the contrary, our approach goes at a lower level, at a more basic logical interpretation that suits for both and then leaves room, we believe, to build up whatever may be appropriate in order to account for semantics and pragmatics of the Web.

Bibliography

F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logics Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

Tim Berners-Lee. The Semantic Web as a language of logic. `http://www.w3.org/DesignIssues/Logic.html`, 1999-2009.

Tim Berners-Lee and Mark Fischetti. *Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. Harper San Francisco, 1999.

Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, May 2001.

Willem Nico Borst. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. PhD thesis, University of Twente, 1997.

Christopher H. Brooks and Nancy Montanez. An analysis of the effectiveness of tagging in blogs. In *AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*. AAAI, 2005.

Michael Genesereth and Nils J. Nilsson. *Logical foundations of Artificial Intelligence*. Morgan Kaufmann Publishers, Los Altos, California, 1987.

Thomas Gruber. Ontology of Folksonomy: A Mash-up of Apples and Oranges. *International Journal On Semantic Web and Information Systems*, 3(1):1–11, 2007. available also online at `http://tomgruber.org/writing/ontology-of-folksonomy.htm`.

Tom Gruber. Toward principles for the design of ontologies used for knowledge sharing. In *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic, 1993. Available as Stanford Knowledge Systems Laboratory Report KSL-93-04.

Tom Gruber. Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human and Computer Studies*, 43(5-6):907–928, November 1995.

Nicola Guarino. Formal Ontology, Conceptual Analysis and Knowledge Representation. *International Journal of Human and Computer Studies*, 43:625–640, 1995.

Nicola Guarino. Formal Ontology and Information Systems. In *1st International Conference on Formal Ontologies in Information Systems, FOIS'98*, pages 3–15. IOS Press, Trento, Italy, 1998.

Nicola Guarino, Daniel Oberle, and Steffen Staab. What is an Ontology? In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, pages 1–17. Springer Berlin Heidelberg, 2nd edition, 2009.

P. Hayes. RDF Semantics. `http://www.w3.org/TR/2004/REC-rdf-mt-20040210/`, 2004.

Domenico Lembo, Carsten Lutz, and Boontawee Suntisrivaraporn. Tasks for Ontology Design and Maintenance (Deliverable 5). Technical report, TONES – Thinking ONtologiES, EU-FP6 project, 2006.

Alexandre Monnin. Qu'est-ce qu'un tag? Entre accès et libellés, l'esquisse d'une caractérisation. In *Actes des 20es Journées Francophones d'Ingénierie des Connaissances "Connaissance et communautés en ligne" (IC2009)*, pages 1–12. 2009.

Natalya Noy and Carole Hafner. State of the art in the Ontology Design. *AI Magazine*, 18(3):53, 1997.

Lévy Pierre. *L'intelligence collective. Pour une anthropologie du cyberespace.* La Découverte, Paris, 1994.

Emanuele Quintarelli. `http://www.iskoi.org/doc/folksonomies.htm`, 2005. ISKO Italy-UniMIB meeting.

Atefeh Sharif. Combining ontology and folksonomy: An Integrated Approach to Knowledge Representation. `http://www.ifla2009satelliteflorence.it/meeting3/program/assets/AtefehSharif.pdf`, 2009. IFLA satellite meetings. Florence, Italy.

Barry Smith and Christopher Welty. FOIS introduction: Ontology – towards a new synthesis. In *Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01, pages .3–.9, New York, NY, USA, 2001. ACM.

Rudi Studer, V. Richard Benjamins, and Dieter Fensel. Knowledge Engineering: Principles and methods. In *Data & Knowledge Engineering*, volume 25, pages 161–197. Elsevier Science, 1998.

Thomas Vander-Wal. Folksonomy. `http://vanderwal.net/folksonomy.html`, 2004.

Thomas Vander-Wal. Explaining and Showing Broad and Narrow Folksonomies. `http://www.personalinfocloud.com/2005/02/explaining_and_.html`, 2005.

C. Welty, F. Lehmann, G. Gruninger, and M. Uschold. Ontology: Expert Systems All Over Again? Invited panel at AAAI-99: The National Conference on Artificial Intelligence, 1999. Austin, Texas.

Part II.

Logic for the Web

The role of Logic for the Web today. And maybe tomorrow

In spite of the large effort to use logic with ontologies in order to exploit inferential capabilities of the "reasoners" – thanks to the use of ontology languages that comply with Description Logics – we focus our attention on a completely different way of using logic for the Semantic Web. That is, we are interested in logic for supporting communication throughout the Web. In particualr, communication of intelligent, autonomous Web agents, i.e. not *ad hoc* designed programs, where the interaction with other agents and the environment is just a detailed rigid protocol to be obeyed. That means to get outside of particular applications, for which a rich and expressive DL ontology, coupled with a powerful inferential engine, can prove extremely useful. For, indeed, there is very little to infer out there (outside of single applications or services) if different ontologies have no viable way to connect with each other. Communication in the Semantic Web indeed, in our view, is not only a matter of syntactical linking between ontologies, but of exchange of information between systems that rely on ontologies written in different languages. And concerning languages, again, it is not a matter of using OWL, possibly in any of its dialects, instead of RDF, or any other standard or non-standard language, but of the "private" languages that are produced with every terminology (i.e. vocabulary, in the acception of vocabulary that we have presented and discussed in the previous part) adopted to describe facts (data, documents, resources ... and their relationships) in the Web.

The first chapter will be devoted to an analysis of the ways to have knowledge and/or information exchanges between automated information systems over the Web by means of Semantic Web technologies. That is, in particular, the processes that to some extent rely on ontologies and possibly are enhanced with the use of Knowledge Engineering techniques. The result will be a sort of survey of these processes whose added value is the attempt to signal how they can be interpreted within a logical frame, alternative to their usual algorithmic account.

The objective of the second chapter, then, will be to find a common ground, with some logical value to be meaningful to machines, where to interpret possibly any form of knowledge representation and / or information description that we may find in the Web – there is actually a variety of alternatives for this, ranging from DL ontologies to folksonomies, passing by (RDF) repositories, open databases (that is, databases made accessible over the Web to external querying services, *via* query mappings), and so on. The aim is clearly to propose such a common ground as a theoretical frame to represent, model, and study how to improve, all the processes that we consider in the chapter just before. We will thus introduce our proposal of Ontological Compatibility Spaces, as a possible candidate for the yet missing layer of unifying logic in the picture of the overall Semantic Web architecture (cf. figure 1.1).

In the last chapter of this part, we will tackle more closely the issue of the interaction between intelligent, autonomous agents. Firstly we will contrast such agents (not yet existing indeed, as well as Semantic Web is not yet in service either) to other web agents that actually exist and are very active on nowadays Web, in an attempt to determine which should be their role, their tasks, their peculiar capabilities and the dynamics of their interaction. And subsequently we will introduce our proposal to describe and model their interaction. To this end we will take advantage of the insights of Ludics and of the innovative approach it brings to Logics to deal with the issue of interaction. But we will also borrow the technical solutions introduced with Ludics, and will slightly re-adapt its formalism to better suit our interest in Semantic Web agents' interaction.

4. Using information over the Web

4.1. Use of ontologies

What do we mean for “use of ontologies”? And why should one use ontologies? As it is embedded in a stand-alone application, even though the application is reachable through internet, an ontology is directly used by the application itself, which is likely to enable, precisely thanks to the ontology, something like a *semantic navigation* or *semantic exploration* of the knowledge that it makes accessible and usable to its users – most likely, indeed, it will be some documental repository.

This is a very basic, even poor, use of ontologies and more generally speaking of so-called semantic technologies, that goes short over HTML capabilities in producing adaptive menus in Web applications. There are much more important reasons and much more interesting ways of using ontologies in the Web. Most of all, we consider as the most important reason to use ontologies the possibility to exchange knowledge and information among different applications and systems. Such an exchange, for example, may happen in a web portal that provides access to data stored in different databases through a system of interface-ontologies, one for each connected database, against which user queries (more or less explicit) are resolved before being fired on the appropriate database – with the patent benefit for the user of not having to find on its own the right database to query, besides not getting involved into the specific query language of that database.

Another significant way of having knowledge exchange facilitated, if not just enabled, thanks to ontologies may be that of a company that acquires another company. The holding company has to save and put together with its own all the knowledge of the acquired company (their data, their accounting documents, their reference manuals, ... all that sums up their information system) and a “layer of semantics” (again say interface-ontologies) would enable appropriate software tools to manage likewise both their original and newly acquired knowledge without having to re-input data, re-register documents ...

Yet another case may be that of scientific reasearch or medical institutes providing tools for the explanation and comparison of their own special terminology with respect to that of other organizations, so as to facilitate the share of scientific results and spread knowledge over the World.

All these examples present different forms of knowledge exchanges, besides the different kinds of knowledge exchanged. They vary also for they attain (or at least aim to) different objectives, diverse purposes that move real world agents (research institutes, industrial and business companies, private individuals, maybe governments too ...) to establish communication relationships, to ask and provide answers, to push and pull information, in general to interact in order to share portions of knowledge, be that for the search of single, punctual information, for the management of eterogeneous amounts of data, or for the understanding of a very hard area of scientific knowledge. Corresponding to such varied examples there is also a great variety in the role that semantic technologies, ontologies in particular, play to enable the exchange. In the following we propose some generalizations out of the actual implementations that adopt ontologies to support interaction between a variety of (real world or virtual world) agents through the Web. We remark by the way that we do not claim for interaction happening between ontologies, which are rather static objects, firm artifacts designed precisely to support interaction. Interaction takes place as the intercourse between agents with some (possibly programmed) intentionality. Ontologies are intended to help agents in mutually understanding what they look for and what they can provide to satisfy others' requests.

In most cases agents are just some virtual extensions of real world agents which are "embedded" in a ready to use web application, so that a user has just to fill in the parameters of a query to get her answer – if they are not precisely human users in front of the screen trying to get aware of the form that has been given to information within the service they are accessing. According to the happiest expectation, however, agents should be(come, one day!) web agents performing quite complex tasks on behalf of their human masters[1]. The closer one gets to such a scenario, the higher must be the level of interaction between automated agents and the more flexible must be the structures that they will use in order to carry their knowledge, since many different agents within a supposedly large and rich world cannot necessarily all share the same specifications for everything. Indeed, an ontology fixes and describes, and possibly also logically defines, the "intended meaning"[2] of types of data that are likely to be found within the domain or application that the ontology is designed to cover, thus organizing structures of types of data. In any ontology, each data type has its own name, so that any two agents that call things (data) with same names should easily agree and succesfully exchange information.

The idea behind web agents interaction is precisely that of a communicative interaction. For the communication to be effective, both agents must know each others' data structure(s), so as to actually "understand" the incoming information. The point is then: how to guarantee that any two agents can mutually understand what they tell? The simplest way would be that they all share the same set of ontologies or, more generally, of logically principled descriptions of data structures, just to be "liberal" as regards the particular model to be used. Or at least they should share a common set of reference ontologies sufficient to frame any other local, domain-specific ontology that could be relevant to some particular application-specific agents. However, this way is not viable at present nor in the short term, and it is reasonable that it could never occur for the whole Web-wide panorama of possible interactions.

Could we get the all-languages vocabulary capable to reveal us – possibly through

[1] Is to be noted that there should be agents also on the part of web service providers (e.g. the web portal of the example above) which will interact with personal agents of private users.

[2] We may say to be short that such intended meaning is the specification on how that data can be used.

a very long series of references – what anyone may say to us in whatever language he is using, including technical jargons, slangs and micro-community languages? Apart from exagerations, the point is that to have such a dynamic for interaction requires to have always previously provided the mapping specifications from a language to any other. This turns out immediately to be a huge cost on the count of feasibility of the overall Semantic Web project, even though in specially limited settings it can work.

What we consider really fascinating, in contrast, is to think of the most general interaction between web agents, one that cannot fully rely on predetermined agreement between the parties on behalf of which the agents act. Therefore we would take, later on, the focus on deeper, more challenging forms of interaction, namely those that imply and cause some reassessment or updating of the knowledge of the agents. Learning agents are absolutely not a novelty, but the way how the updating should happen based on the logical frame that we will propose wants to be quite a novelty.

Now let's go back to the difficulties that appear when agents speak different languages, that is adopt different ontologies. In this case it needs to discover some basic points on which to establish interaction. The traditional approach requires the work of human experts to discover possible agreements between the two ontologies and then formalize the necessary translation rules[3]. Most (all in practice) of the ways of use of ontologies that we are going to talk about are performed according to this approach. According to it, the basic task is to find correspondences between the languages. This can be seen at the level of the language itself by measuring similarity of the terms used to denote data types, which is something that machines can do very well. However, more interesting and more significant correpondences are to be found at the level of meaning. Talking of meaning brings us in the realm of semantics, where the troubles of Semantic Web live. Indeed we should first of all distinguish at least two different orders of meaning: one that is formal semantics, that can be treated according to Model Theory and provides the interpretation of the formal theory "packed" in any ontology; and one that is what the man on the street thinks of meaning, with no subtle distinction about sense, denotation and the similar, something that we would call generally, before entering any deeper discussion, "real world semantics" as opposite to formal semantics.

As soon as we are committed with "real world semantics", we have also to rely on both common sense and specialists' knowledge about the particular domain with which the ontology is concerned. Then, since common sense and specialists' knowledge require obviously human intervention, we can recognize quite clearly the bottleneck that retards the achievement of a real Web-wide interaction: it needs ready-made "interaction protocols" for any possible interaction, and most likely such protocols require human work to be defined – unless one is willing web agents performing actions that do not agree with real world nature and states of affairs.

The morals, for the time being, is that we are not able to tackle interaction between any two agents that do not speak the same language. We cannot even conceive forms of minimal interaction for this case: if they do not speak the same language and if their

[3] Although more interesting should be the possibility to let the agents really interact on the basis of their knowledge and find on their own possible correspondences.

ontologies do not refer to some common ontology[4] they cannot interact at all. Normally all the options (syntactical similarity, model theoretic interpretation and "real world semantics" correspondences) are taken into account in working with ontologies to support knowledge exchange, and a sort of collaboration between humans and machines occurs, with the latter helping the former in finding possible correspondences by signalling syntactical similarities and verifying models – thanks to inferential engines.

4.2. Ontology of ontologies

Before we can discuss the ways of using ontologies, we must consider on which parts of an ontology one can operate. As we said in the first part of this work, ontologies are usually conceived as made of two main components, namely T-box and A-box. This is a convenience that allows to keep separate (or separable) two different perspectives on the domain that an ontology deals with, and whose significance is related to the reading of ontologies as First Order theories. An ontology, indeed, is a theory built up of the axioms that it contains in the T-box and, where available, in the A-box. The (logical) language of such a theory is made of the standard terms defined in the specification of the particular language used to design the ontology (e.g. the usual OWL) – which provides the logic operators available in the language for a DL- (or FOL-) compliant ontology language – plus the terms introduced in the ontology itself, that is predicate-symbols (concepts and relations) and individual-names. To be more precise, given an ontology $\mathcal{O}$ we may distinguish within the alphabet $\mathcal{A}(\mathcal{O})$ of the language $\mathfrak{L}(\mathcal{O})$, in which the ontology is specified, two distinct sets of symbols: the set $\mathcal{L}(\mathcal{O})$ which contains all the logical symbols – that is in particular the set of logical operators admitted for the particular Description Logic adopted, in case of a DL-compliant ontology – ; and the signature $\mathcal{S}(\mathcal{O})$, that is the set of all the non logical symbols of the language. In particular, the signature for a DL ontology complying with the W3C standards shall contain all the predicate names used to describe the domain of the ontology, thus involving symbols for unary (the concepts) and binary (the roles and properties, in general: relations) predicates. For an ontology given together with an A-box, the signature shall comprise also the set $Ind(\mathcal{O})$ of the individual names that appear in the A-box. Such a set will be the set of individual constants for the language $\mathfrak{L}(\mathcal{O})$ of the whole ontology. With the full language $\mathfrak{L}(\mathcal{O})$ an ontology is actually produced in the form of a (First Order) theory, whose axioms may be divided into two sets which correspond precisely to the T-box and A-box. The distinction between T-box and A-box axioms depends on the presence of individual names in the formula: normally A-box assertions are precisely all and only those in which some individual appears[5].

In the following we will disregard in general the distinction between the terms proper to the specification of the ontology language chosen, that is the logical terms of $\mathcal{L}(\mathcal{O})$, and the signature $\mathcal{S}(\mathcal{O})$ which actually bears what is really peculiar to an ontology.

[4]Besides those that specify the language for ontologies itself, e.g. OWL, that's clear.

[5]In quite exceptional cases, individual names may appear also in the T-box – where they are called nominals – but that is not important to our discourse.

Indeed, when comparing the languages of two ontologies, unless it is specified otherwise, we will always be focusing just on their signatures, since $\mathcal{L}(\mathcal{O})$ can be assumed to be always the same. To say it all, in the panorama of Description Logics there are a number of variants of the same basic language which differ each from the other ones for the stock of logical operators that they admit. But, on the one hand, actually the ontologies for Semantic Web may be defined in a very limited set of DLs (basically just three are implemented in W3C standards) and the issues concerning their reciprocal translations, as a matter concerning the possible loss of expressivity in translating from one to another, have been largely investigated in DLs focused researches and are not of our interest here; on the other hand, Semantic Web ontologies may even be defined in formalisms which have no correspondence with DLs and, possibly, even no simple interpretation as First Order theories – in rdf for instance the same term could act both as a predicate symbol and as an individual name. So we prefer to consider as the general case, and then care more about it, the differences in languages that do not depend especially on the different formalisms adopted, but on the different representations of reality that are embodied into the ontologies. Indeed, this is (much more than mismatches in logical formalisms) the kind of language differences and communication difficulties that we would like to face and (help to) resolve as regards information exchanges over the Web. After all, we are going to propose a logical representation suitable for ontologies as well as folksonomies, which have *per se* no interesting First Order representation…

Having said that an ontology provides a representation of some piece of reality, we can observe that such a representation is produced combining the specific terms of the signature $\mathcal{S}(\mathcal{O})$ by the logical operators provided with the formal language $\mathcal{L}(\mathcal{O})$. In particular, if we restrict to the subset of unary and binary predicates in $\mathcal{S}(\mathcal{O})$ we get what is often called a *vocabulary*. The combination of the elements of the vocabulary by logical operators produces logical formulas, which are the axioms of the theory that specifies the ontology.

Now, the interest in having separated T-box and A-box in ontologies gets explained: as every theory, an ontology with its axioms restricts the set of models satisfying the language. The axioms in the T-box (that is concept definitions and predicate specifications) cause a restriction of that set that still does not get down at the level of specific states of affairs. The models satisfying a *terminology* (as T-boxes are sometimes called) are modelled according to constraints that holds for every possible state of affairs. On the contrary, the axioms in the A-box introduce a particular state of affairs, yet leaving room for multiple (infinite) models. It is clearly not a matter of cardinality of the sets of models available for the theory. It is however a matter of detail, granularity of the "world" that the ontology describes, which is, by the way, the semantics of the application, service or whatever it is, that the ontology supports.

The distinction between T-box and A-box however does not call for another "kind of meaning" as we were talking above, since the logical meaning of both terminologies and "full ontologies" – i.e. those which present also a particular state of affairs into some A-box, which are indeed knowledge bases – deals always with nothing but model theoretical semantics, with just the difference that the support set is fixed for KBs. Even though it is not restricted to any particular state of affairs, T-boxes logical interpretation does not reach the other order of meaning, what we called above "real world semantics", which is the most natural one and the only that grounds the intended meaning of an ontology within the picture of external reality – which in turn is partially reflected into the ontology when the designer produces it. That order of "meaning" however is the place for authentic ontology (or at least deeper, more aware of philosophical speculation over the centuries), where e.g. also primitive concepts of a particular ontology get somehow defined in terms of other concepts, relations and logical operators that express truly ontological considerations and which cannot (and are not to) be expressed with the same language of T-boxes[6].

We may now sum up with the following schematic list what is to be retained from the discourse above in order to safely follow our reading of the processes of use of ontologies. An ontology $\mathcal{O}$ is specified by a theory $\mathcal{T}(\mathcal{O})$ written in a language $\mathfrak{L}(\mathcal{O})$. The alphabet $\mathcal{A}(\mathcal{O})$ of such a language is a couple consisting of

- the logical formalism $\mathcal{L}(\mathcal{O})$ adopted (e.g. some Description Logic)
- and the signature $\mathcal{S}(\mathcal{O})$, which in its turn consists of
  - a set $Voc(\mathcal{O})$ of unary and binary predicate symbols, which will be respectively concept names and relation (or role) names, that is the *vocabulary* of $\mathcal{O}$;
  - and a set $Ind(\mathcal{O})$, possibly empty, of individual constants, which will be individual names in the A-box.

Some remarks:

- In DL ontologies the sets $Voc(\mathcal{O})$ and $Ind(\mathcal{O})$ are disjoint, whereas OWL-Full and RDF do not pose this constraint. We respect the constraint as far as we talk about DL ontologies.
- The set of axioms written in $\mathfrak{L}(\mathcal{O})$ in which are used only predicate symbols from $\mathcal{S}(\mathcal{O})$ (i.e. no individual names) forms a T-box.
- Axioms in a T-box typically provide definitions for concepts and relations.
- For relations in particular the axioms specify between which concepts they hold.
- The set of axioms in which also always appear individual names is an A-box.
- A T-box together with an A-box produce a Knowledge Base (KB).

4.3. "Operations" on ontologies

There is remarkable confusion about how to call many different "operations" that are performed on ontologies with different purposes, so that it is quite difficult, if

[6]Some researchers, for example, have proposed to adopt modal logic operators for this part of ontology specification ([Guarino et al., 1994]).

not impossible, to provide a universally acceptable classification – and by the way we may note that W3C has never proposed a set of standard operations or processes on ontologies, as they do not look like caring about that. First of all, it should be clear that we do not aim to talk here about the process of ontology generation. Rather we are interested in how ontologies may support interaction over the Web, so that we care for those processes on ontologies that cause knowledge, information and data to be shared among and exchanged between different organizations, at any level. We will sometimes use the term "operation" as a general word to call all of such processes on ontologies, although it is apparent that in most cases they are not operations in a logical/mathematical sense. In fact, they often require a sort of handcraft work, though assisted with semi-automatic techniques, so that the definitions that one may find in the literature usually refer to some specific algorithm and suggest a "creative" component in order to keep the resulting ontologies sounding. On the contrary, within our effort to provide a general logical account of the use of ontologies for Semantic Web, we aim to define real operations whenever it is possible; otherwise to state the specific relation that holds between ontologies that are the result of some particular process of use of ontologies. We base, then, our analysis of the operations and, more generally, of the processes of composition and/or comparison of ontologies on the summarizations provided in [Grau et al., 2006] and in [Marek, 2006], but we do not adhere too strictly to either.

Let us begin our presentation of "operations" on ontologies with a quite simple process and a fundamental relation between ontologies, which are necessary to adequately account for the other more complex processes and operations.

Ontology Segmentation – Modularity Ontology Segmentation is the process that cuts an ontology $\mathcal{O}'$ out of an ontology $\mathcal{O}$. It is not really an operation, since it gives no unique result. Indeed one could extract many different "sub-ontologies" $\mathcal{O}'$, $\mathcal{O}''$, ... from the same ontology $\mathcal{O}$. The special request for ontology segmentation is that the extracted ontology $\mathcal{O}'$ preserves all the information that depends on the terms that it uses, i.e. in particular all the axioms from $\mathcal{T}(\mathcal{O})$ relevant to the sub-signature $\mathcal{S}' \subseteq \mathcal{S}(\mathcal{O})$ that is used for the extracted ontology $\mathcal{O}'$. Indeed, ontology segmentation is produced by selecting a sub-language from that of the original ontology $\mathcal{O}$ and taking the sub-theory made of all the axioms in $\mathcal{O}$ where are used the terms and symbols appearing in the sub-language chosen[7]. We talked about a sub-language and noted it just $\mathcal{S}' \subseteq \mathcal{S}(\mathcal{O})$ because, as we said in the previous chapter, we are interested in particular in the vocabulary of the languages, whereas changes in the logical formalism, affecting the selection of logical operators available to formulate axioms, touches the technical area of conversions between Description Logics dialects. By the way, we think that this part can be very interesting for knowledge engineers who have to compose "semantic services", but should be of minor interest for an agent looking for information

[7] Or *viceversa*, and more likely, one may first choose a set of interesting axioms to be extracted and then recover the relevant sub-language, with respect to which even other axioms could need to be included into selection in order to preserve the intended meaning of the terms as they are defined in $\mathcal{O}$.

over the Web, since it has to find and retain information rather than convert it in more or less expressive formal languages. Therefore we simply assume that the logical formalism $\mathcal{L}(\mathcal{O}')$ is the same as $\mathcal{L}(\mathcal{O})$. Nevertheless it costs nothing to consider $\mathfrak{L}(\mathcal{O}') \subseteq \mathfrak{L}(\mathcal{O})$, which clearly holds for $\mathcal{S}(\mathcal{O})$ is part of $\mathfrak{L}(\mathcal{O})$.

As [Ghilardi et al., 2006] put it out, this implies that $\mathcal{O}'$ is a *uniform interpolant* of $\mathcal{O}$, thus establishing a peculiar relation between $\mathcal{O}$ and $\mathcal{O}'$ as regards the possibility of use of the information that they contain. Indeed, for every T-box or A-box formula A containing only terms that are both in $\mathcal{O}$ and $\mathcal{O}'$

$$\mathcal{O} \text{ ontology}; \quad \mathcal{O}' \subseteq \mathcal{O} \quad \text{with} \quad \mathcal{O} \vdash A \quad \Leftrightarrow \quad \mathcal{O}' \vdash A$$

The extracted ontology $\mathcal{O}'$ may be thought of as a *module* of $\mathcal{O}$, that is a "detachable" component of $\mathcal{O}$, such that it can be reused in another ontology (see below **Inheritance**).

We may better define the relation holding between the original ontology $\mathcal{O}$ and the extracted ontology $\mathcal{O}'$ precisely by means of the notion of module.

$$\begin{aligned} \mathcal{O}' \text{ module of } \mathcal{O} \quad \Leftrightarrow \quad & \mathfrak{L}(\mathcal{O}') \subseteq \mathfrak{L}(\mathcal{O}) \\ \wedge \quad & \mathcal{T}(\mathcal{O}') = \mathcal{T}(\mathcal{O}) \text{ restricted to } \mathfrak{L}(\mathcal{O}') \end{aligned}$$

where to restrict a theory, e.g. $\mathcal{T}(\mathcal{O})$, to a language which is a sub-language of the original language of that theory means to form a theory, like $\mathcal{T}(\mathcal{O}')$, that contains all and only the formulas of $\mathcal{T}(\mathcal{O})$ that belong to the sub-language. In the following we may note the restriction as $\mathcal{T}(\mathcal{O}) \restriction \mathfrak{L}$.

We observe that any module of an ontology produced by ontology segmentation is a homogeneus module, that is the extracted part is yet an ontology (a sub-ontology), provided that the signature of the module is not empty. In particular it must be $Voc(\mathcal{O}') \neq \emptyset$. In this way, it may be noted, one could in principle have an ontology made of just one predicate symbol, with a trivial defining axiom, as the result of a segmentation. Technically it may happen, indeed, since there is no special criterion to decide below which threshold an ontology does no longer worth the name of ontology. Anyway, the point of our remark about *modularity* of ontologies is that any sub-language is in principle able to represent a sub-ontology, but typically interesting (useful) modules (i.e. sub-ontologies) are the ones that are produced by just taking a reduced vocabulary $Voc(\mathcal{O}') \subset Voc(\mathcal{O})$, discarding the set $Ind(\mathcal{O})$ – so that the A-box too gets discarded – and preserving the same logical formalism $\mathcal{L}(\mathcal{O})$, so as to preserve the same expressive power as $\mathcal{O}$.

A final remark about ontology segmentation is that it induces a partial order between the original ontology and its possible modules. Just to give the basic idea, this order can be seen immediately as an ordering over the width of the signature of each ontology.

Ontology Extension – Inheritance Ontology Extension is, in a good approximation, the inverse of segmentation since it causes the enrichment of the language of the ontology $\mathcal{O}$, and presumably also of its theory. One may see this happen as soon as a new module is added to a pre-existing ontology $\mathcal{O}$, thus producing the new ontology $\mathcal{O}'$. Indeed, from the point of view of $\mathcal{O}$, its language gets extended and its theory presumably too. Inheritance, then, is the relation holding between the two ontologies as it can be seen from the opposite point of view, that of the new ontology $\mathcal{O}'$ which is said to inherit from the pre-existing ontology $\mathcal{O}$.

More generally, inheritance comes into play whenever an ontology $\mathcal{O}'$ is designed reusing, even only partially, another ontology $\mathcal{O}$, which may be quite a frequent case with Semantic Web ontologies. It suffices, indeed, that ontology $\mathcal{O}'$ imports (cf. section 2.2.2, part I) ontology $\mathcal{O}$. In this latter case however one usually needs not import the whole ontology $\mathcal{O}$, but a subset of its language, together with the relevant axioms, that contains all the terms that will be used in $\mathcal{O}'$. In other words: it is enough to reuse just a module of $\mathcal{O}'$. Such a subset of the language and theory of $\mathcal{O}$ (the imported module) must act as a uniform interpolant of $\mathcal{O}$, so that we have here clearly a case of ontology segmentation seen from the opposite point of view. Anyway, for the general case we may keep on considering the whole ontology $\mathcal{O}$. Then, in the practice of Semantic Web ontologies, $\mathcal{O}'$ is said to be an *extension* of $\mathcal{O}$ when $\mathcal{O}'$ is specially designed to extend the knowledge in $\mathcal{O}$. This is exactly the same as in mathematics, where a theory may be extended for some special purpose with a set of new axioms and – even though one is most likely to really use only a subset of the original theory which is extended – the extension s/he produces should stay compatible with the full original theory. Thus also in Semantic Web, when an ontology imports another one, the importing one should agree with everything that is expressed in the imported one. Given the linkable nature of OWL and RDF ontologies (thanks to URIs and the namespace technique), together with the expectation to actually have a web of interconnected ontologies, one can see how inheritance is (or should be) one of the most basic elements of Semantic Web architecture.

A typical case of having such a use of an ontology is for "recycling" of already available and possibly largely accepted ontologies or for framing some application-specific ontology within the larger world described by a foundational ontology. Another remarkably typical case is that of coupling an A-box to a T-box (terminology). Now it is clear what is the profit from having distinct "boxes": A-boxes can be interchangeably and independently plugged to one single terminology, and in the case of Semantic Web ontologies it is enough to provide a link to the terminology, with no need to have a local copy of it.

Even though usually the knowledge that one wants to inherit from an external ontology is limited to its terminology (that is T-box formulas), we may all the same consider our ontologies as wholes made of a T-box as well as an A-box: it causes no loss of generality to our discourse (quite the contrary in case) – just keep in mind that importing an A-box implies accepting also a specific state of affairs beside a particular description of a domain. As regards the inferences that can be drawn in $\mathcal{O}'$ with respect to $\mathcal{O}$, and then the use of ontology $\mathcal{O}$ through $\mathcal{O}'$ – that proposes an interesting setting of interaction over ontologies for web agents – we have that, being $\mathcal{O}'$ an extension of $\mathcal{O}$

$$\mathcal{O} \vdash A \quad \Rightarrow \quad \mathcal{O}' \vdash A$$

where A is any T-box or A-box formula containing only terms that appear also in $\mathcal{O}$.

We then briefly define as follows the relation holding between an ontology $\mathcal{O}'$ inheriting from $\mathcal{O}$:

$$\begin{array}{rcl} \mathcal{O}' \text{ inherits from } \mathcal{O} & \Leftrightarrow & \exists \mathfrak{L}' \subset \mathfrak{L}(\mathcal{O}') \text{ s.t. } \mathfrak{L}' \subseteq \mathfrak{L}(\mathcal{O}) \ \wedge \ \mathfrak{L}' \neq \emptyset \\ & \wedge & \mathcal{T}(\mathcal{O}) \restriction \mathfrak{L}' \ \subseteq \ \mathcal{T}(\mathcal{O}') \\ & \text{so that} & \forall A \in \mathfrak{L}' \quad \mathcal{O} \vdash A \ \Rightarrow \ \mathcal{O}' \vdash A \end{array}$$

Besides, inheritance too induces a partial order between ontologies, but a more significant one than that induced by segmentation. Indeed, for inheritance one can go back to a deeper, more basical, definition of concepts.

What is most interesting about this way of using ontologies, is that it shows precisely the same behaviour of the *cut rule* in sequent calculus.To put it in a nutshell, it is the basic idea behind the use of a lemma in the logical demonstration of a theorem. Inheritance establishes a connection between two theories, like a *cut* does between two proofs, that is two logical demonstrations of logical statements. Every proof, indeed, has its premises and its conclusion. If one considers the proof as a computation (a process that derives inferences in fact), the premises are the expected input for the process and the conclusion is its output. A cut takes place where a proof produces as output that which another proof expects as its input, so that the two proofs can be "plugged" so as to compose a single proof with the premises of the first one and the conclusion of the other. By the way, this is *communication* in Logics, and we may see here the sharing of information. And we can imagine that the same mechanism will trigger web agents communication on finding a match (e.g. a common term) between their ontologies. This would let them share other parts of their ontologies beside the matched term, namely all that is involved with that term: the terminological axiom that defines it, and those in which it appears. Inheritance, like a cut, allows to move from one to the other theory, and then possibly to go back up to the proper axioms of the other ontology (or theory, or proof) to find the reason why some consequence can be derived and held.

Now let's tackle some processes and operations a little more difficult to describe in a general but logically strict setting. We start with the mapping of ontologies, which is the core of a number of other operations and which looks also to be the enabling mechanism of a working Semantic Web – it is not by chance that it is the most studied operation by the way. Indeed, if one considers the tasks that web agents should accomplish in Semantic Web, the greatest challenge is to have solid and reliable strategies to allow an agent to compare and position some "new" concept that it comes to meet with respect to the conceptual structure that it already knows – what should roughly imitate that which happens in our minds when we catch a new concept. Just after the mapping we introduce a series of other operations that realize the mapping with additional constraints or under special conditions.

Ontology Mapping The mapping process is at the core of many forms of data integration, even independently of ontologies. Nevertheless, ontologies are powerful tools to support data integration. We will try here to focus especially on the mapping as a process that affects directly ontologies, since ontologies themselves may "contain" data liable to integration and also the mere terminologies of ontologies can be worth of a mapping. Then, to map an ontology $\mathcal{O}$ into another ontology $\mathcal{O}'$ is the process that immerses (part of) the knowledge of $\mathcal{O}$ into $\mathcal{O}'$ by establishing some "links" from $\mathcal{O}$ to $\mathcal{O}'$. This quite general discoursive definition needs first of all to be refined according to different aspects of the knowledge that ontologies contain and that can be mapped. In a first, quite simple approximation – that focuses only on the technical aspect – an ontology mapping is just a particular kind of data mapping (semantic data mapping), where the terms used to name types of data are changed. Such terms appear as concepts in an ontology and are used to label (tag) semantically annotated data. It is basically a renaming operation that provides one-to-one translation of terms from a system to another. The purpose here is to access data contained within a system, described according to an ontology, through a service or application that uses its own ontology to communicate with other systems. Obviously, it is not always possible to have a corresponding term in $\mathcal{O}'$ for any term in $\mathcal{O}$. What happens to data missing a suitable translation depends on what the service or application asking for the mapping must obtain – probably those data will be simply disregarded and discarded.

Links between the two ontologies expressing such kind of mapping are simple logical assertions like $\forall x(P(x) \leftrightarrow P'(x))$, where P and P' are predicate symbols of $\mathcal{O}$ and $\mathcal{O}'$ respectively. Thanks to such rules data available in $\mathcal{O}$ as of type P become accessible in $\mathcal{O}'$ as of type P' – the interest is essentially in the possibility to call things (data) with the other name, P' instead of P. The same translation rule can be expressed as a declarative specification, and possibly integrated within an ontology instead of being recorded as an application-specific rule. In such a case it will be an axiom like $P \equiv P'$.

Really, it would be enough – and it is usually done this way – to have the translation in only one direction, the interesting one according to the specific setting of the application that uses those ontologies. Then the axiom can be something more similar to $\forall x(P(x) \rightarrow P'(x))$. The same can be registered directly in one of the ontologies, namely the one that looks into the data of the other, in our case $\mathcal{O}$, as a concept inclusion axiom of the form $P \sqsubseteq P'$. By the way, once a mapping is produced thanks to this kind of axioms, we have again inheritance come into play, for a piece of the language of $\mathcal{O}'$ is borrowed by $\mathcal{O}$ and a connection, a *cut* is produced between the two ontologies.

Anyway, it is not always only a matter of (formal) languages to be translated.

Along with a vocabulary of terms for semantic data annotation[8], an ontology provides a knowledge representation, which means the description of some domain from a specific point of view. To map an ontology into another one then implies also to deal with issues like verify whether the domain is really the same, whether the two ontologies actually overlap in describing it and also confrontating the intended meaning expressed through the axioms. In addition, the complexity factor still increases if any or even worse both of the ontologies come with the A-box, that means to get involved also with particular states of affairs. It is therefore really hard (and very unlikely) to have a full mapping of one ontology into another one (see below ontology refinement).

Now let's consider in more details the different forms in which a mapping between ontologies may occur. In a first case we have mapping between terminologies. In a second case we may face a mapping between a terminology and a full ontology, a knowledge base, that is T-box plus A-box. In a third case we may look for a mapping between two knowledge bases.

1. The mapping between terminologies is perhaps the most studied as a task related to ontologies, and it is surely the most stimulating. It is at the boundary between information systems, linguistics and philosophy [Guarino, 1998]. Here researchers are concerned with how to discover when two ontologies describe the same world, though using different languages and depicting that world in quite different ways – each one according to a particular point of view on that world, since inevitably any designer must adopt one. Here then KR experts or ontologists have to perform a sort of mediation between the two terminologies. Such a mediation may require and pass through many steps like: exploration of the linguistic meanings of the terms used in each terminology; computation and/or understanding of the intended meaning, both through modeltheoretical interpretation and relevant human knowledge; discovery of coincident and of divergent concepts; negotiation concerning the parts of the original ontologies, and their intended meaning, liable to be preserved in the mapping. (Formal) logic has little to say about this kind of mapping since most of the tasks focus on aspects of the ontologies that cannot be effectively dealt with by means of any inferential calculus. In fact, (formal) logic intervenes as far as the check of models satisfaying the ontologies is concerned. In particular, logic comes into play to check the formal correctness of the result of the mapping, also with respect to the original ontologies. That is surely an important role, but (formal) logic does not affect, unless marginally, the way how the mapping is found, it alone cannot lead to the discovery of the mapping.

   This kind of mapping, once established, is recorded by just writing the equivalence or subsumption axioms between terms from the two ontologies.

[8] This is a peculiar facility provided by an ontology especially when it is a Semantic Web ontology, but it is not mandatory. There are many ontologies which are not conceived nor used for such a purpose. Just to have an idea one may think of foundational ontologies which are intended to frame the intended meaning of other ontologies.

For example, an equivalence axiom $C \in \mathcal{O} \equiv C' \in \mathcal{O}'$ is to say that a concept C in ontology $\mathcal{O}$ corresponds, and then can be translated, to concept C' in ontology $\mathcal{O}'$ and *vice versa*. On the contrary, an axiom like $C \in \mathcal{O} \sqsubseteq C' \in \mathcal{O}'$ will allow for translation in just one direction. It indeed does not state a full correspondence, but indicates that concept C' “contains” concept C – or C implies C' – which can be an acceptable workaround to link a piece of knowledge between $\mathcal{O}$ and $\mathcal{O}'$ if one cannot do anything better. Even though in practice most of the links between two terminologies are not bidirectional, but unidirectional, as in the last example, nevertheless the axioms may be much more complex and meaningful than those that we have just seen. Precisely for the lack of perfect correspondents, a concept C of $\mathcal{O}$ may be approximated with terms of $\mathcal{O}'$ through a (re-)definition. Just to give a basic example, our concept C could be found to be equivalent to a concept not yet existing in $\mathcal{O}'$ but such that can be described through a logical formula: $C = C' \wedge C''$ with both C' and C'' concepts of $\mathcal{O}'$. No need to say that relations (and properties) too may undergo the same treatment.

Generally, we may describe the mapping as a function that, once the single mappings of concepts and relations have been provided, associates to a predicate symbol of $\mathcal{O}$ a logical formula that uses the language of $\mathcal{O}'$ and that defines a (possibly new) concept or relation. It can be presented as the partial function $map : \mathcal{O} \rightarrow \mathcal{O}'$.

Such a mapping affects just the intensional level of ontologies producing a translation from a concept of an ontology $\mathcal{O}$ to a concept of another ontology $\mathcal{O}'$. To be more precise, the mapping takes as input a term of $\mathcal{O}$ and leads to: i) a term in $\mathcal{O}'$ which denotes the same concept (or relation) where available; ii) or to a concept (or relation) of $\mathcal{O}'$ such that the first implies this latter; iii) or to a “fresh” concept (or relation) newly defined using terms and predicate symbols of $\mathcal{O}'$. Here, by the way, is hidden the assumption that is possible to identify with each other two concepts expressed with different terms in different languages by different groups or individuals that adopt different points of view even on the same small portion of the world – though it is to be verified also that it is really the same portion of world. That is not generally incorrect, but it is a delicate issue that requires a careful and philosophically aware ontological discernment.

2. The mapping between a terminology and a knowledge base looks very like the general case of semantic data mapping between databases. Indeed, the basic idea is that of an A-box that gets accessed through a terminology that is different from “its own” – since its own is precisely the one with which the A-box complies “by construction”, i.e. simply because it is written using the terms (predicate symbols) of that terminology. The point is then to establish which terms of the new, external terminology will (re)cover individuals (and their “facts” recorded in the A-box) according to the original terminology.

   With this kind of mapping it is expected to produce a sound inter-connection of data sources over the Web. It derives directly from the practice with databases and still preserves from its origins an implicit assumption about the scenery that frames the use of information. That is, there must be some organization that "owns" data and provides access to data through a more or less complex network of mapped ontologies. Such a scenario is known also under the name of mediator-based data integration [Calvanese et al., 2006], for which the relevant literature counts two specific and alternative architectures called *global as view* (GAV) and *local as view* (LAV) [Lenzerini, 2002]. The two approaches concern the way how mappings are specified and both produce mappings for the concepts (and for roles and properties too) as query rewriting, as it is used with databases. Nevertheless – since ontologies can be put on top of databases as interfaces (through another specific mapping that we may consider here as "hard-coded" and not relevant to our discourse) and since queries over ontologies are written as logical formulas – precisely the same approaches hold also for us and we recall them for the general case of mappings between ontologies. The difference between GAV and LAV regards the direction of rewritings. GAV requires that a view, i.e. a query, over the data sources is associated with every concept of a global ontology that covers the whole knowledge of the inter-connected data sources; whereas LAV requires the sources to be defined as views over the global schema. In the field of data integration also other approaches do exist, e.g. GLAV (Global/Local as view) [Lenzerini, 2002] and BAV (Both As View) [McBrien and Poulovassilis, 2007], the former being a variant of LAV that allows to use queries to build a global schema, and the latter being a quite complex way of expressing each schema as a query over the other system's schema.

Anyway, whichever be the architecture used we are always faced to the usual "prepared Semantic Web" where the necessary mappings must be produced in advance by experts and may hold, obviously, only with respect to the terminologies that they have taken into consideration in designing the complex service.

Just to give an idea, let's imagine that our ontology $\mathcal{O}$ is the global ontology of a large web portal collecting different data sources. In case the connections among them are provided by mappings specified according to the GAV model, for each unary predicate symbol $P \in \mathcal{O}$ (expressing a concept of $\mathcal{O}$) and for each binary predicate symbol $R \in \mathcal{O}$ (expressing a relation between concepts of $\mathcal{O}$), we should have a query Q_i over some local ontology $\mathcal{O}'$ such that

$$\begin{array}{rrcl} \text{for } P \text{ (unary)} & P(x) & = & Q_j(x) \\ \text{for } R \text{ (binary)} & R(x,y) & = & Q_k(x,y) \end{array}$$

since indeed the query Q is nothing but a logical formula in which appear only terms and predicate symbols proper to the local ontology $\mathcal{O}'$, and where the values for which the variable x (and the pair x, y) is to be checked are data (individual names and/or property values) from the same local ontology $\mathcal{O}'$. Then, to a concept (or role or property) of $\mathcal{O}$ corresponds a rule, a logical formula that makes a reasoner to compute individuals (and facts) collectable under that predicate symbol. Since we are dealing with ontologies, the logical formula underlying the query can be expressed directly as a concept (or role or property) definition[9] This way, the mapping can be specified through the same kind of axioms of which a T-box is normally made of. Thus, in the simplest case the concept expressed by P may be defined in the global ontology $\mathcal{O}$ as the set $\{x \mid P'(x)\}$ with $P' \in \mathcal{O}'$ whereas a relation $R \in \mathcal{O}$ as the set $\{(x, y) \mid R'(x, y)\}$ with $R' \in \mathcal{O}'$. But much more complex formulas (queries) can be used as concept definitions in the global ontology. For example, an atomic concept, expressed by means of a unary predicate P in $\mathcal{O}$ may be realized as the set $\{x \mid P'(x) \wedge R'(x, p)\}$, always with P' and $R' \in \mathcal{O}'$ and p a constant (e.g. individual name) of $\mathcal{O}'$, i.e. as a query with a constraint over the relation R' of the local ontology $\mathcal{O}'$. Yet more interesting is the case of a concept in the global ontology that is realized drawing data from different sources, i.e. two distinct local ontologies, thus operating in the setting of a mapping that involves three distinct ontologies: the global one ($\mathcal{O}$) and two locals ($\mathcal{O}'$ and $\mathcal{O}''$). For example, the set $\{(x, y) \mid R'(x, y) \vee R''(x, y)\}$ may be the realization of a relation R defined in the global ontology $\mathcal{O}$ as the union of the relations R' and R'' of the two distinct local ontologies $\mathcal{O}'$ and $\mathcal{O}''$ respectively.

We can see behind all this again a mapping function, with something more. Indeed, once the global ontology is given – that is to say once the hard work has been carried out by some human experts who have registered all the necessary inter-definitions as queries over local ontologies – we have both a mapping function like that of the previous case (the mapping between terminologies) that affects concepts, relations and the like, and also a more "concrete" process that injects data (individual names, values, resources) from local sources, that we look at through the local ontologies $\mathcal{O}'$, $\mathcal{O}''$ ... into the global repository, that we look at through the global ontology $\mathcal{O}$.

Therefore we can recognize a double movement, in opposite directions: the *map* partial function that, as before, operates from the (global) ontology $\mathcal{O}$ to local ontologies $\mathcal{O}'$, $\mathcal{O}''$ ... associating queries (logical formulas) that use the language of local ontologies to predicate symbols of $\mathcal{O}$; and the process that we could call *realization*, that brings data from local sources (accessed through the local ontologies) to the global repository, seen through the global ontology. We may present this process, similarly to a function, as $rea : \mathbf{S_1}, \ldots, \mathbf{S_n} \to \mathbf{R}$, where $\mathbf{S_1}, \ldots, \mathbf{S_n}$ are the local sources of data (e.g. single A-boxes) and $\mathbf{R}$ is the global repository, in which sets of data, corresponding to the concepts and relations of the global ontology $\mathcal{O}$ are

[9] At least in most cases. Exceptions may depend on a mismatch between the complexity of the query and the expressivity of the Description Logic used for the ontology.

stored[10]. As for the *map* function, the *rea* process too is partial: it is defined for the recorded views (i.e. queries) over the local sources, that is the views that are also definitions for predicate symbols in the global ontology $\mathcal{O}$.

A final remark regards consistency of the global ontology with respect to local ontologies, and of the global repository with respect to the local sources. As for the previous case of mapping between two terminologies, the consistency and soundness of the ontology used to access data (the global ontology) depends on the skills of the KR and/or domain experts that carry out the task, providing the mappings for every predicate to be realized. Consistency check can be performed thanks to reasoners. Additional constraints can also be introduced, to better model the domain or possibly to guarantee soundness of data gathered and put together from different local sources – but the behaviour of the application (e.g. our web portal) in presence of such additional constraints, i.e. whether to filter data according to them (correctness, with respect to the global ontology) or to retrieve all data available (completeness, with respect to local sources), is a matter of *policy* that goes beyond our interest here.

3. The mapping between two knowledge bases is usually considered at present just in research applications. In particular it is considered together with the particular setting of a network of data sources, each one with its own ontology, that share information. Such a setting is sometimes called *peer-to-peer* (P2P) *network of ontologies* ([Calvanese et al., 2006]). This is undoubtedly a very interesting setting in the perspective of Semantic Web, even though for the time being the process of discovery of the mappings relies, here too, on experts's knowledge, so that it is, once again, rooted in the need for given in advance mapping rules, and is not a really dynamical setting of interaction between any two peers. However, this network-of-peers scenario stays of great interest and quite close to our idea of interaction, where web agents not only share knowledge but also exchange (and collect from others) punctual information, that is A-box facts.

   Now we can imagine immediately a new order of problems: what does it imply to "mix up" individuals from different sources? Is it licit, correct and sound? Indeed, whereas in the previous case, the mapping between a data source and a (different) terminology used to access data, the transfer of data is possible in only one direction, within a network of peers each link between two nodes can be run in both directions. Nevertheless, in spite of the apparent increase of complexity in the scenario, nothing happens that is really different with respect to the previous case. Indeed, communication among peers can be obtained according to two approaches: to adopt a common language for all the peers with respect to which are defined mappings for every peer's ontology, or to have one-to-one mappings for any pair of

[10] Actually in real applications of mediator-based semantic data integration, the concepts and relations of the global schema may be *realized*, that is the data copied from the local sources and the generated views recorded, in order to speed up the querying process.

peers. It is easy to see that the first approach is a centralized one, for it requires a sort of global ontology, and we are back to a mediator-based application or service scenario. Then we must consider just the "distributed" approach, that requires a specific mapping for every pair of peers.

As regards the issue about mixing data from many different sources, we have already met it, though with no special attention, while talking about the mediator-based mapping between a knowledge base and a terminology. Indeed, under the umbrella of a global ontology $\mathcal{O}$, may be realized concepts or relations – in any case realized as sets of data at the extensional level – composed of data drawn from many local sources. It was the case with our simple example about a relation $R \in \mathcal{O}$ that is mapped on two distinct relations R' and R'' from two distinct local ontologies, and that is realized at the level of data through the union of the data collected under R' and R''. All this happens safely, on the basis of the mapping specifications provided. The important issue about the soundness, correctness and significance of the mixed data depends uniquely on the consistency and soundness of the world description embedded in the ontology $\mathcal{O}$. Precisely the same holds when looking at a network of peer ontologies: the possible realization of a concept, within the local source of any single peer, with the composition of internal data and external data drawn from other peers, is as significant, correct and sound as the mapping specification is with respect to the intended meaning of relevant predicates in both the ontologies involved. Nevertheless, it is actually less easy to verify the adequacy of the mappings all over the network. And also a little harder is to describe the mapping (and the realization) as a function.

Before speaking of the adequacy of the mappings all over the network we should spend a few words about the architecture of peer-networks of ontologies. Indeed, it is assumed ([Calvanese et al., 2006]) that the peer network works as follows: any peer that receives a query both provides an answer based on its own local data – here comprised also data drawn from other sources and added to internal ones as the result of realization operated at a given time (that which involve an updating issue) – and also routes the request to other peers that it knows, thanks to mapping specifications, that are able to answer the same query. That is, basically, it refers to the peers towards which it has a mapping specification for the predicates that it uses to answer the query. Now, given such a mechanism, one can consider the risk for distortion and corruption of the query, like in the child game "chinese whispers" (or "telephone game") – and then also for progressive fall of relevance of the answers retrieved from the far off regions of the network. It is a sort of "semantic noise" that could cause, for a query passing throughout the network, the appearance of divergent results due to "long distance misunderstandings". That is, slight shift in the intended meaning from a peer to the next one which are not important over the first few passages, but on the long distance of many passages may cause major changes.

Anyway, that is a risk that scales easily at the level of the whole Semantic Web as soon as one thinks of Semantic Web according to the same distributed approach that we have seen for peer networks of ontologies. By the way, regarding such networks some strategies have already been proposed to tackle the problem of the progressive meaning shift [Calvanese et al., 2006]. But since it is not our focus for this work, we close here the point, waiting for the time to return on it for a few considerations from our alternative point of view on Semantic Web.

The second difficulty that we noted above is about how to recognize the mapping as a function in the network scenario. Indeed we have no more a single ontology (the global one) whose language is mapped onto the languages of a set of other ontologies (the local ones). Then we have just to consider one-to-one mappings, as in fact they should be constructed by KR experts (and data owners) who weave the peer network by writing the mapping specifications.

We can now draw our conclusions about the mapping as a function operating at the intensional level of ontologies (that is on terminologies, T-boxes), with a corresponding process, though working in the opposite direction, operating at the extensional level (facts in knowledge bases, A-boxes).

The *map* function operates over terminologies, taking as input a predicate symbol (concept, relation or property) and returning, in the general case, a logical formula; on the contrary, the *rea* process operates over data sources (sticking to ontologies we can say KBs) taking as input sets of data and returning, in the general case, their union. Indeed, *map* and *rea* may yield, respectively:

- *map* directly to a predicate name: when the mapping specification is like $map(P(x)) = P'(x)$, for $P \in \mathcal{O}$ and $P' \in \mathcal{O}'$ – that means that in the destination ontology there is already a predicate compatible with the one in the departure ontology – and *rea* leads to the neat transfer of a set of data from a source to another ($rea(X \subseteq \mathbf{S}) = X \subseteq \mathbf{S}'$);
- *map* to an explicit logical formula: when such a formula is the definition of a predicate that does not yet exist in the destination ontology, since in this latter there is no ready-to-use predicate symbol equivalent to $P \in \mathcal{O}$, but nevertheless a predicate that can be described, and defined, using predicates (and possibly terms) of $\mathcal{O}'$. In such a case *rea* actually operates as a complex query (it could be a conjunctive query, including join of views and selections based on given parameters) like the ones shown as example for the case of mapping between a terminology and a knowledge base. It retrieves, according to the query specification, the set X of data out of the data stored in the local source and transfers them into the global repository (mediator-based case), or another local source (P2P network case).

The connection between *map* and *rea* is the mapping specification for any single predicate, that is essentially the logical formula, the query itself, that transforms a predicate symbol into a set of data.

The notion of equivalence that underpins the *map* function is quite informal, because it depends on the assessment of similarity of the intended meaning delimited by the definitions given with term descriptions within the terminologies (the system of axiom contained in the ontology T-box). Therefore, it is an equivalence modulo approximation of the intended meaning of an ontology with respect to the other(s) to which it is mapped.

But in this way we are still too close to the usual accounts of the mapping that deal with the special application context in which the mapping is performed. To have a really general definition of the mapping we propose the following (remember what we said about the language of ontologies in 4.2).

Mapping – general version A mapping from ontology $\mathcal{O}$ to ontology $\mathcal{O}'$ is a partial function ϕ from the vocabulary of $\mathcal{O}$ into the (full) language of $\mathcal{O}'$, i.e. $\phi : Voc(\mathcal{O}) \to \mathfrak{L}(\mathcal{O}')$, such that:

- if ϕ is defined for a unary predicate symbol (a concept name) $P \in Voc(\mathcal{O})$ then $\phi(P)$ is a monadic formula (i.e. a formula that contains only one free individual variable) of $\mathfrak{L}(\mathcal{O}')$;
- if ϕ is defined for a binary predicate symbol (a relation name) $R \in Voc(\mathcal{O})$ then $\phi(R)$ is a dyadic formula (i.e. a formula that contains exactly two free individual variables) of $\mathfrak{L}(\mathcal{O}')$.

As a consequence, we get a new theory $\mathcal{T}(\mathcal{O}) \restriction \mathfrak{L}(\mathcal{O}') = \{A \mid \phi(A) \text{ is defined}\}$. Now, if the mapping has to affect not only terminologies, but knowledge bases (i.e. ontologies with A-box), we have that:

- for the simplest case where ontology $\mathcal{O}$ is just a terminology used to access data in $\mathcal{O}'$ nothing else is needed and

  $$\forall P \in Voc(\mathcal{O}) \text{ s.t. } \phi(P) \text{ is defined } \{x \mid P(x)\} \subseteq \{x \mid \phi(P)(x)\}$$

  $$\forall R \in Voc(\mathcal{O}) \text{ s.t. } \phi(R) \text{ is defined } \{\langle x, y\rangle \mid R(x,y)\} \subseteq \{\langle x, y\rangle \mid \phi(R)(x,y)\}$$

  for respectively P unary and R binary predicate symbols. Note that the use of $\subseteq$ depends on the possibility to have loose mappings, i.e. the ones in the style of $P(x) \to P'(x)$ instead of $P(x) \leftrightarrow P'(x)$;
- for more complex situations where also the mapped ontology $\mathcal{O}$ has its own A-box, we have to recover also the original set of individual names from the signature $\mathcal{S}(\mathcal{O})$, which indeed is the pair $\langle Voc(\mathcal{O}), Ind(\mathcal{O})\rangle$. Therefore we have to slightly accomodate the definition above so as to have available together with the re-writing of $Voc(\mathcal{O})$ in $\mathfrak{L}(\mathcal{O}')$ also the set of actual data contained in $\mathcal{O}$. That which we can get by extending the language $\mathfrak{L}(\mathcal{O}')$ with the set of individual constants from $\mathfrak{L}(\mathcal{O})$, that is $Ind(\mathcal{O})$.

Therefore our final comprehensive definition for the mapping will be

$$\phi : Voc(\mathcal{O}) \to \mathfrak{L}^{\mathcal{O} \to \mathcal{O}'}$$

with $\mathfrak{L}^{\mathcal{O}\to\mathcal{O}'} = \mathfrak{L}(\mathcal{O}') \uplus Ind(\mathcal{O})$. In particular, with OWL (or generally speaking Semantic Web) ontologies, which use the namespaces technique, one needs just the simpler $\mathfrak{L}(\mathcal{O}') \cup Ind(\mathcal{O})$ since individuals appearing in the A-boxes would be distinguished based on their namespace – and in the remote case where some individual would appear in both A-boxes, it should actually be interpreted as the same individual. In particular we observe that

$$\mathcal{T}(\mathcal{O}) \vdash A \quad \Rightarrow \quad \mathcal{T}(\mathcal{O}) \restriction \mathfrak{L}^{\mathcal{O}\to\mathcal{O}'} \vdash \phi(A)$$

whenever $\vdash$ is a recursive predicate, as it is in DL-compliant ontology languages like OWL-Lite, OWL-DL and OWL2.

Final and perhaps the most important remark about the mapping is that it provides the new theory $\mathcal{T}(\mathcal{O}) \restriction \mathfrak{L}^{\mathcal{O}\to\mathcal{O}'}$ which is a direct extension of $\mathcal{O}'$, and also of $\mathcal{O}$ *via* the translation produced by the function ϕ – it is easy to see that the conditions expressed in the definition of extension / inheritance are satisfied between $\mathcal{T}(\mathcal{O}) \restriction \mathfrak{L}^{\mathcal{O}\to\mathcal{O}'}$ and $\mathcal{T}(\mathcal{O}')$, and between $\mathcal{T}(\mathcal{O}) \restriction \mathfrak{L}^{\mathcal{O}\to\mathcal{O}'}$ and $\mathcal{T}(\mathcal{O})$. That means that the ontologies $\mathcal{O}$ and $\mathcal{O}'$ are bridged in such a way that it is possible to have logical *cuts* between them.

Such an account for the mapping operation is more general and, irrespective of any special condition under which the mapping is performed, observes the basic process of finding data in an ontology that satisfy the definition of concepts or relations of another ontology. In particular, this definition for the mapping i) considers only the two ontologies $\mathcal{O}$ and $\mathcal{O}'$ while making no assumption about a third ontology (like the *global ontology* of a mediator-based systems) which possibly stores the links between the ontologies in the form of queries embedded in predicate definitions; ii) it requires to talk about just one function ϕ that operates on predicate names and affects corresponding data; and iii) highlights the fact that the mapping (as well as the next three mapping-based "operations" that we are going to define) is an operation on languages, which is as necessary as we are committed to, and reliant on, linguistic description and definition of knowledge in order to share it.

Ontology Refinement It may look like a special case of mapping, or an original operation in between ontology mapping and the mere evidence of inheritance, for it requires that all the predicate symbols in an ontology $\mathcal{O}$ have an equivalent in another ontology $\mathcal{O}'$. But refinement does not provide links between ontologies, rather it provides a one-way (unidirectional) full translation of an ontology into the other one, though it happens obviously through the (re-)definition of predicate symbols of $\mathcal{O}$ in the language of $\mathcal{O}'$. Refinement then, unlike mapping, is a total "operation": it must provide translation for the whole ontology $\mathcal{O}$. The refining aspect indeed is that also primitive predicate names (i.e. those for which is given no logical definition) in ontology $\mathcal{O}$ are translated into non-primitive, i.e. logically defined concepts, in the ontology $\mathcal{O}'$. That is to say that the representation of the domain with which $\mathcal{O}$ is concerned gets somewhat more articulated in $\mathcal{O}'$.

It is quite clear then that the real interest of such an operation is limited to the case of a knowledge exchange between terminologies, with no data (A-box facts) passing from a source to the other. And equally clear is that refinement may succeed only under the very special case where the ontology that gets extended (our ontology $\mathcal{O}$) is already fully conceptually contained in the other ontology (our $\mathcal{O}'$)[11]. By the way, the easiest way to have that – and the one that actually is followed – is to design the extending ontology purposedly to refine the pre-existing ontology $\mathcal{O}$.

Precisely for the way how refinement is performed this operation gets very close to be a mere manifestation of the relation of inheritance. Refinement thus produces an ontology with which the pre-existing ontology can be entirely *cut*: the primitive terms in $Voc(\mathcal{O})$ get defined based on the terms of $\mathcal{O}'$, so that one can rewrite the whole ontology $\mathcal{O}$ using the language of $\mathcal{O}'$. We can see that in a way very similar to the mapping process, through a function which is total for this time, $\psi : Voc(\mathcal{O}) \to \mathfrak{L}(\mathcal{O}')$ such that for every unary (resp. binary) predicate P of $Voc(\mathcal{O})$, $\psi(P)$ is a monadic (resp. dyadic) formula of $\mathfrak{L}(\mathcal{O}')$.

Now, let us call $\mathcal{O}_{\mathfrak{L}(\mathcal{O}')}$ the rewritten version of $\mathcal{O}$. We have that $\mathcal{O}_{\mathfrak{L}(\mathcal{O}')}$ inherits $\mathcal{O}'$ – because in $\mathcal{O}_{\mathfrak{L}(\mathcal{O}')}$ are used terms imported from $\mathfrak{L}(\mathcal{O}')$; and $\mathcal{T}(\mathcal{O})$ gets transformed into $\mathcal{T}(\mathcal{O}) \restriction \mathfrak{L}(\mathcal{O}')$ which is an extension of $\mathcal{O}'$. Indeed, terms of $\mathfrak{L}(\mathcal{O}')$ and the axioms defining them in $\mathcal{T}(\mathcal{O}')$ are used as building blocks to produce $\mathcal{T}(\mathcal{O}) \restriction \mathfrak{L}(\mathcal{O}')$. As a consequence we find that also ontology refinement induces a partial order between ontologies – and it is the more informative one that we have met so far (precisely that of inheritance): $\mathcal{T}(\mathcal{O}') < \mathcal{T}(\mathcal{O}) \restriction \mathfrak{L}(\mathcal{O}')$ and also $\mathcal{T}(\mathcal{O}) < \mathcal{T}(\mathcal{O}) \restriction \mathfrak{L}(\mathcal{O}')$ *via* the translation of $Voc(\mathcal{O})$ produced with the refinement.

One should note in particular that the refinement works in only one direction: all the terms in $Voc(\mathcal{O})$ get (re)defined in $\mathfrak{L}(\mathcal{O}')$, but no assumption can be done in general for the terms in $Voc(\mathcal{O}')$ along the inverse direction. In particular, “new” primitive terms in $Voc(\mathcal{O}')$ will have no correspondent in $\mathfrak{L}(\mathcal{O})$.

Ontology Alignment It is a bidirectional mapping where two ontologies are bridged to have, at least in principle, information flowing from each other. As for the mapping, each link between the two ontologies is unidirectional, but the existence of outgoing links from both the ontologies causes the bidirectionality of the overall process. Alignment should provide translation of (part of) the knowledge of each ontology into the language of the other one, with the permission for “marginal” loss of information during the process – i.e. it is expected that not everything can be successfully translated. In the practice, alignment can be seen typically as the form of composition of a (narrower) domain ontology with a (very large and broader) foundational ontology, where attention is focused primarily on terminologies (T-boxes) in order to have the broad foundational ontology framing

[11] Since such a condition is quite heavy, in the practice refinement may occur for just a sub-ontology, a module, of $\mathcal{O}$. However that does not require special accomodation of the general presentation we are giving of ontology refinement.

the more specific and constraining domain ontology. In this case then there is usually little (if anything) that can be drawn from the specific domain ontology to the foundational one – since the links are intended just to bring the finer ontological setting designed in the foundational ontology to the narrow domain ontology. The operation therefore may look quite the same as one of the forms of mapping that we have discussed above. But a significant difference depends on the purpose for which the operation is performed: the ultimate aim for mapping is to access data (facts), whereas alignment is intended to represent knowledge in an exhaustive and sound manner though limiting the effort in designing it by relying on other ready-to-use ontologies – possibly ontologies quite largely accepted so as to improve also shareability, to ease spreading of the narrow domain ontology. In fact, beside the case of the alignment concerning a specific domain and a foundational ontology, the general case for alignment involves ontologies that are complementary as regards the domains that they are to describe.

Anyway, for the general case we have to consider links departing from both ontologies. The difference with respect to mapping is then precisely the reciprocity of the operation, although it is not a full "mirroring" between the two ontologies (cf. ontology unification below) since mappings do not affect ontologies in their whole, but just parts. Then we recover the definition of mapping and accomodate it in order to have room for two functions operating in the opposite directions: an alignment between ontologies $\mathcal{O}$ and $\mathcal{O}'$ is a pair of partial functions

$$\phi_1 : Voc(\mathcal{O}) \to \mathfrak{L}(\mathcal{O}')$$
$$\phi_2 : Voc(\mathcal{O}') \to \mathfrak{L}(\mathcal{O})$$

such that, for P predicate symbol in $Voc(\mathcal{O})$ on which ϕ_1 is defined, $\phi_1(P)$ is a formula of the language of $\mathfrak{L}(\mathcal{O}')$, and the set of the data of type P is included, or the same as, the set of data retrieved by the "query" $\phi(P)$, that is $\{x \mid P(x)\} \subseteq \{x \mid \phi_1(P)(x)\}$; and the like for $P' \in Voc(\mathcal{O}')$: if ϕ_2 is defined on it then $\phi_2(P')$ is a formula in $\mathfrak{L}(\mathcal{O})$ and the corresponding set of data $\{x \mid P'(x)\} \subseteq \{x \mid \phi_2(P')(x)\}$. All that we said previously for the mapping holds here too, in this two-ways setting, for the alignment. In particular, the two ontologies get bridged and logical *cuts* may take place, *via* translations of the languages provided by ϕ_1 and ϕ_2.

Ontology Unification The relation holding between aligned ontologies in general does not allow for a reciprocal translation of *all* the terms of $Voc(\mathcal{O})$ and $Voc(\mathcal{O}')$. In order to find such a complete reciprocal translation one should perform ontology unification. As the refinement is a more artificious version of the mapping – since it requires a second ontology specially designed to comply with and extend the first one – the unification may be considered as a more artificious version of the alignment, since it requires that the two ontologies involved propose fully overlapping conceptualizations, though in different languages, so that one should only translate terms with terms (or formulas) from a language to the other. The purpose of ontology unification, indeed, is to make inference in an ontology mappable into inference in the other ontology, or to say it in a simpler way, the objective is to arrange two ontologies in such a way that they are exactly equivalent as regards logical consequences that can be derived based on their axiomatization. Such a relationship between two ontologies is interesting from the point of view of a possible improvement with respect to the complexity of the reasoning in one or the other ontology, depending on the hardness of concept definitions in either language. This is possible iff given the two functions $\psi_1 : Voc(\mathcal{O}) \rightarrow \mathfrak{L}(\mathcal{O}')$ and $\psi_2 : Voc(\mathcal{O}') \rightarrow \mathfrak{L}(\mathcal{O})$

$$\begin{array}{rlll} & \forall P & \in Voc(\mathcal{O}) & \psi_1(P) \text{ is defined} \\ \text{and} & \forall Q & \in Voc(\mathcal{O}') & \psi_2(Q) \text{ is defined} \end{array}$$

That is to say that for every (unary or binary) predicate symbol P in the vocabulary of either ontology it is possible to recover an equivalent formula expressed in the language of the other ontology, thanks to one out of the two total functions ψ_1 and ψ_2. In particular, as regards the possibility to map inference from an ontology to the other one we remark that, for any T-box or A-box formula A_i

$$\begin{array}{rlcl} & \mathcal{T}(\mathcal{O}) \vdash A_j & \Rightarrow & \mathcal{T}(\mathcal{O}') \vdash \psi_1(A_j) \\ \text{and} & \mathcal{T}(\mathcal{O}') \vdash A_k & \Rightarrow & \mathcal{T}(\mathcal{O}) \vdash \psi_2(A_k) \end{array}$$

As regards the partial order among ontologies, either of two ontologies that suit to unification cannot precede the other, unsurprisingly since they are essentially the same ontology.

Finally we conclude with another series of operations concerning the integration of (possibly many) distinct ontologies into a new, different, larger ontology. Even though some people consider the *ontology integration* itself as a specific operation among the others that we will name just below – in a fashion that makes it difficult to finely distinguish among them all –, we prefer nonetheless to keep ontology integration as a general term for a few different approaches and techniques. We may distinguish these operations with respect to the above series of mapping-based operations precisely because the following ones produce in every case new *per se* meaningful ontologies which gather the knowledge from the single original ontologies – still preserving their intended meaning – whereas the mapping-based processes aim to unveil correspondences and establish connections between (parts of) two or more already existing ontologies in order to access knowledge from each other – and only incidentally it may appear some “new” ontology, but just as the place where one stores the links (or translation rules) that are established.

In order to cumulate the knowledge available in different ontologies one has to face the question about the compatibility of the ontologies to be integrated. In the general case any two different ontologies may not be compatible, i.e. may yield into contradiction. So we can distinguish about two basically different ways of integrating ontologies, based on how this issue is faced. On the one hand we find a form of integration that cares only about formal correctness of the result, that is a purely syntactical integration that guarantees for the resulting ontology not yielding into contradiction, but that, on the contrary, does not affect the "intended meaning" of the involved ontologies – since it deals with a quite artificious notion of interpretation. On the other hand we have a variety of forms of integration that primarily focus on the accuracy of the intended meaning brought by the resulting ontology with respect to that of the original ones. Nonetheless in this latter case the process of integration may become (and usually does) "incomplete", that is it causes to discard part of the constraints posed to interpretation by the single ontologies in favour of a weaker but larger overall interpretation in the larger ontology.

Ontology Union The simplest (but weakest) way of integrating two ontologies is therefore to do it by embracing the first way out of the two that we have just mentioned (syntactical correctness). More precisely, given any two ontologies $\mathcal{O}$ and $\mathcal{O}'$, consider the ontology $\mathcal{O}''$ which is the sum of the knowledge of both $\mathcal{O}$ and $\mathcal{O}'$ – a sort of union whose value, from the point of view of ontologists, may look like just a logical workaround, but which will prove useful in our subsequent proposal for representing Web knowledge.

In introducing such a "syntactical union" of ontologies, intended as a merely logical operation, we refer to (and somehow paraphrase) the notion of "controlled union of ontologies", which produces a *conservative extension* of both the original ontologies, as [Ghilardi et al., 2006] and [Grau et al., 2007] put it.

Therefore we will resort to the set theoretical operation of disjoint union, which is the key for the safe handling of the whole ontologies (T- and A-boxes). Indeed, in order to preserve the sets of legal and "safe" inferences that can be drawn *from any of the original ontologies* $\mathcal{O}$ and $\mathcal{O}'$ – which is one of two possible readings of "preserving their intended meaning" – we need just to colour the ontologies. That is enough to make us sure that the union will not trigger off any contradiction and that everything that can be proved within each ontology alone still can be proved within the union – we get trivially the sum of the legal inferences from both the original ontologies, and nothing more. The colouring will be easily obtained through the renaming of the signatures of $\mathcal{O}$ and $\mathcal{O}'$, that is the logical procedure of changing the names of the predicative and individual constants appearing in the signatures so as to have no same symbol occurring in both[12].

Now, assuming that $\mathcal{O}$ and $\mathcal{O}'$ alone are consistent, we have that they are surely consistent even together. We have not to check for the condition that ontologies

[12] We hold that renaming preserves the intended meaning in such a syntactical reading of the matter. Imagine to prefix all symbols of the signatures with a unique prefix for each ontology. It is easy to see in such an operation the same facility that namespace policy (largely used in Semantic Web) offers in order to refer to the intended meaning of any particular term according to its "owner". And by the way, from the merely logical point of view, the renaming procedure is licit since predicative and individual constants are reasonably constants as far as one is committed to the habits of First Order languages, but their very nature, visible at Second Order, is that of variables.

$\mathcal{O}$ and $\mathcal{O}'$ together do not yield into contradiction, since compatibility comes "for free" after renaming. It is a consequence of the Craig Interpolation Theorem [Chang and Keisler, 1976], for which if a theory T proves some formula F, then there exists a formula F' such that $T \vdash F'$ and $F' \vdash F$ and, most of all, every predicate or constant symbol (excluding identity) occurring in F' occurs also both in T and F.

In order to demonstrate this general guarantee for compatibility of the merged ontologies, let us consider the ontologies as First Order theories, with O and O' as conjunctions of the axioms in $\mathcal{O}$ and $\mathcal{O}'$ respectively. The point is that it is no longer possible to build a Craig's interpolant F of $O \wedge O'$ that yields into contradiction. Indeed, if $O \wedge O' \vdash \perp$, for the Craig Interpolation Theorem we would have a formula F such that $O \wedge O' \vdash F$ and $F \vdash \perp$ and in which occur only predicate and constant symbols that are also in all O, O' and $\perp$. From that it is already apparent that the only common symbol is $\perp$, so that the interpolant must be $\perp$ itself, but this way we gain no new information and force to check again the consistency of $O \wedge O'$. Anyway we may formulate the possible contradiction of $\mathcal{O}$ and $\mathcal{O}'$ in the form of $O \vdash \neg O'$. Then, the interpolant F would play thus: $O \vdash F$ and $F, O' \vdash \perp$. Now, after renaming, the only common symbols in $\mathcal{O}$ and $\mathcal{O}'$ are the logical constants for true (1) and false ($\perp$), and we have two possibilities: either $F = 1$ or $F = \perp$. If $F = 1$, we have $1, O' \vdash \perp$ that contradicts the original hypothesis of $\mathcal{O}'$ consistent; if $F = \perp$, we have $O \vdash \perp$ that contradicts the original hypothesis of $\mathcal{O}$ consistent too. Hence, $O \wedge O'$ cannot entail $\perp$. Finally, we can briefly present the union of two ontologies $\mathcal{O}$ and $\mathcal{O}'$ as

$$\mathcal{O}'' = \mathcal{O} \uplus \mathcal{O}'$$

a formulation that subtly shows our interest for the ontologies not as vocabularies but as "containers" of information.

Whenever the ontologies to be integrated are Semantic Web ontologies defined in OWL (or RDF) we have that no special renaming is needed since the namespace technique guarantees unicity of names. Nevertheless, precisely because of the namespaces and the import mechanism (cf. p. 48) between ontologies, we may consider the special case of a union between ontologies that import from other ontologies. What deserves attention here is just the case where the import may cause the ontologies to overlap on some part, i.e. they use same terms (exactly the same, from the same namespace) – otherwise we may yet take the two languages as distinct. This happens when i) one ontology imports from the other, and the union involves both; ii) or both the ontologies import from yet another ontology. In the first case it is not really a union: it is rather the case of an ontology extension (about which we have said above). In the second case, assuming that both the importing ontologies are conservative extensions of the third, imported ontology, no contradiction may depend on the shared part of the signature in the final union of ontologies, as far as it stays unmodified. Let $\mathfrak{L}_1$ and $\mathfrak{L}_2$ be the languages of the importing ontologies $\mathcal{O}$ and $\mathcal{O}'$ respectively, and $\mathfrak{L}_3$ the (portion of) language of another ontology, the imported (part of an) ontology. Note that in case of import between ontologies, the imported language will contain only T-box formulas (no A-box formula) – this depends just on the practical usefulness of importing (parts of) terminologies to be (re-)used over one's own data, whereas typically nobody requires to import data *via* such a mechanism. Now, the Craig Interpolation theorem tells us that the two importing ontologies would become incompatible (contradictory together) only if, among the new axioms introduced in $\mathcal{O}$ and $\mathcal{O}'$ wherein are used terms of $\mathfrak{L}_3$ there are axioms that, once put together in the unified ontology, are contradictory, and the contradiction depends exclusively on terms from $\mathfrak{L}_3$, since beyond $\mathfrak{L}_3$ the two languages $\mathfrak{L}_1$ and $\mathfrak{L}_2$ are clearly disjoint and independent of each other. To check that such a condition is not met is not at all trivial: it would require a consistency check of the whole unified ontology. And, on the other hand, it would be pretty reductive of the usefulness of the import mechanism to consider any form of constraint by which the imported language $\mathfrak{L}_3$ should be developed in the importing ontologies in some "safe mode" so as to avoid such contradictions – the simplest and most debasing constraint might be for instance to keep freezed the imported $\mathfrak{L}_3$ and its relevant (terminological) axioms, by introducing neither in $\mathcal{O}$ nor in $\mathcal{O}'$ any (new) axiom involving terms of $\mathfrak{L}_3$.

If we do not accept to take into consideration any such constraint, we have just two possibilities: i) to keep on with the idea of union of ontologies as a disjoint union, possibly by forcing some colouring even in the case of Semantic Web ontologies when the import mechanism produces overlapping parts between the two ontologies; ii) or to call for some other operation, namely an *ontology merging* that, operating by multiple steps, can take apart the overlapping parts, so as to take them into adequate consideration, and then proceed by means of disjoint union.

Ontology Merging With the name of ontology merging we refer to the process that follows another possible reading of "preserving the intended meaning of the original ontologies". That is to find and put together in a new single ontology (or logical theory), those parts of the two ontologies that builds up a single consistent ontology, i.e. one that does not yield into contradiction – but not necessarily keeping all the legal inferences that can be drawn from each original ontology alone – and possibly allowing for new ones. In any case, no previously legal inference gets contradicted; but some inference may be no longer provable, due to the loss of "marginal" information. This means that the merging of ontologies emphasizes the possible overlappings of the ontologies, in spite of incompatible parts which are to be identified and "fixed" or simply cut off. But that also requires to really understand what the ontologies describe and to perform additional designing tasks in order to identify and distinguish overlapping and expendable parts, and then tailor a new ontology which carries the most and most important of both the original ontologies.

We accept then under the "umbrella" of ontology merging the large variety of special algorithms that, based on different techniques (natural language processing, data/text mining, ...), focus on the optimization of the integration task intended as a human-driven computer-aided process, such that the mechanizeable part – what machines can do for this task – is to help human experts (KR specialists) in understanding the intended meaning of each ontology and to suggest possible mappings between two ontologies in order to design a merged ontology that respects the interpretations of overlapping parts.

It is apparent in such a setting that a strictly logical account of the merging is somewhat inconvenient, whereas a good presentation of such a process requires specific human-mind capabilities to be called into play. Given two ontologies $\mathcal{O}$ and $\mathcal{O}'$, their merge is an ontology $\mathcal{O}''$ that should preserve all the knowledge originally in $\mathcal{O}$ and $\mathcal{O}'$. Nonetheless, the request for conservativeness is not so "mandatory" nor reflected in any special commitment largely recognized ([Grau et al., 2006]) so that actually, in order to guarantee consistency of $\mathcal{O}''$, any (or both) of $\mathcal{O}$ and $\mathcal{O}'$ may be modified. Indeed the common idea behind the general process of ontology merging is that of a join of the theories (usually just the terminologies, T-boxes) underlying ontologies $\mathcal{O}$ and $\mathcal{O}'$ once the axioms that could raise some inconsistencies have been pruned off. This way, the merge actually regards sub-ontologies purposedly extracted from $\mathcal{O}$ and $\mathcal{O}'$, whereas their extraction is a case of ontology segmentation. There is no special requirement concerning how to extract suitable sub-ontologies apart from the usual one: the selected sets of axioms from the ontologies involved in the merging must be such that their join is consistent – and we are again in the vicious circle of moving the problem one step on with no really clear answer.

Anyway, the join of such sub-ontologies is not a union (sum) as in the previous case, since overlapping predicates are to be taken into account in the merging process, highlighted and exploited. This is the reason for the reliance on the special techniques that we quickly mentioned above (such as NLP, data mining, ...), which are adopted precisely to unveil when such overlappings are met – even if it is not the same predicate symbols that are used to denote the same "intended predicate" in two different ontologies – and to signal them to the KR expert in charge of the merge.

Whichever be the composition of techniques adopted in a merging strategy described in the literature till now, we always find solutions that are effective with respect to either a particular language (or class of languages) of Description Logics or to specific constraints that the resulting ontology must respect in order to comply with the particular system that will use it. Anyway, both the vagueness of the requests in the available descriptions of this operation and the ingeneering approach that focuses on implementations have made the name of ontology merging quite well known, but the actual operation quite confuse. Moreover, the production of the sub-ontologies from $\mathcal{O}$ and $\mathcal{O}'$ is a delicate process usually performed through *ad hoc* adjustments and without any general rule to obey – not to say about the confused request that asks for preservation of the knowledge in the original ontologies and just after cheats by allowing for altered ontologies (cut, reduced, simplified ontologies).

As a consequence, as regards our definition of ontology merging, we can just sketch some steps along which two ontologies should get merged, based on the "operations" that we have defined till now. Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be two ontologies to be merged:

- The first step that one can consider is an alignment of $\mathcal{O}_1$ and $\mathcal{O}_2$ which leads to a pair of theories $\mathcal{T}(\mathcal{O}_1) \upharpoonright \mathfrak{L}(\mathcal{O}_2)$ and $\mathcal{T}(\mathcal{O}_2) \upharpoonright \mathfrak{L}(\mathcal{O}_1)$ that express part of the knowledge contained in each ontology with the language of the other one. This step is needed in order to acknowledge and profitably exploit the (possible) overlapping parts of $\mathcal{O}_1$ and $\mathcal{O}_2$.
- A second step, which is really the hardest part to describe in a logical manner, consists in a careful work by knowledge engineers and domain experts in order to "soften" and harmonize concept (and role) definitions produced in $\mathcal{T}(\mathcal{O}_1) \upharpoonright \mathfrak{L}(\mathcal{O}_2)$ and $\mathcal{T}(\mathcal{O}_2) \upharpoonright \mathfrak{L}(\mathcal{O}_1)$, in such a way that it is possible to develop a common language $\mathfrak{L}_0$ in which can be rewritten the axioms generated for the alignment between $\mathcal{O}_1$ and $\mathcal{O}_2$. This part could be thought of as the refinement of each of the provisional theories appeared from the alignment into a new, temporary ontology $\mathcal{O}_0$, with $\mathcal{T}(\mathcal{O}_0) = \{A_j \mid \phi_1(A_j) \text{ is defined}\} \cup \{A_k \mid \phi_2(A_k) \text{ is defined}\}$ [13]:

$$\mathcal{T}(\mathcal{O}_0) \;\subseteq\; (\mathcal{T}(\mathcal{O}_1) \upharpoonright \mathfrak{L}(\mathcal{O}_2) \;\cup\; \mathcal{T}(\mathcal{O}_2) \upharpoonright \mathfrak{L}(\mathcal{O}_1))$$

  via the refinement of the sublanguages $\mathfrak{L}(\mathcal{O}'_2) \subseteq \mathfrak{L}(\mathcal{O}_2)$ in $\mathfrak{L}_0$ and $\mathfrak{L}(\mathcal{O}'_1) \subseteq \mathfrak{L}(\mathcal{O}_1)$ in $\mathfrak{L}_0$. That is, the sublanguages actually used to map predicate symbols of, respectively, $Voc(\mathcal{O}_1)$ and $Voc(\mathcal{O}_2)$ during the previous step of alignment are fully translated into the new language $\mathfrak{L}_0$. In brief, this step provides the basic elements of that which will be the language proper to the finally resulting merged ontology. It is during this step that some "deformations" of the original intended meaning may occur, and it depends on the knowledge about the specific domains and on the expertise of knowledge engineers who care for this process to make them minimal.
- The third step consists of the extension of $\mathcal{O}_0$ by the parts of $\mathcal{O}_1$ and $\mathcal{O}_2$ that remain after the alignment – i.e. the sub-ontologies $\mathcal{O}''_1 \subset \mathcal{O}_1$ and $\mathcal{O}''_2 \subset \mathcal{O}_2$ for which no mapping has been provided by the alignment – into the provisional $\mathcal{O}_0$. That can be quite easily obtained by "pumping" the language $\mathfrak{L}_0$ by adding all the terms from $\mathcal{S}(\mathcal{O}_1)$ and $\mathcal{S}(\mathcal{O}_2)$ that had not been involved in the alignment – and the like with the axioms in $\mathcal{O}_1$ and $\mathcal{O}_2$ unused for the alignment. This step is needed to recover all the knowledge from the original ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ that does not overlap and finally provides us with the resulting merged ontology $\mathcal{O}_3$, specified in the language $\mathfrak{L}_3 = \mathfrak{L}_0 \cup \mathcal{S}(\mathcal{O}''_1) \cup \mathcal{S}(\mathcal{O}''_2)$ and whose axioms form the theory $\mathcal{T}(\mathcal{O}_3) = \mathcal{T}(\mathcal{O}_0) \cup \mathcal{T}(\mathcal{O}''_1) \cup \mathcal{T}(\mathcal{O}''_2)$.

The result of the merging is a new single *per se* meaningful ontology, whereas alignment keeps the two ontologies distinct but "connected" *via* a third ontology that

[13] ϕ_1 and ϕ_2 are the mapping functions that operate the alignment.

stores the links between them (cf. [Noy and Musen, 1999]). By the way, we must note that our choice about showing a new ontology $\mathcal{O}''$ (or global ontology as in the case of mediator-based data mapping) as the place where links between mapped (refined, aligned) ontologies are stored somehow shadows such a difference between merging and alignment. Indeed, it is common habit to consider the construction of a new ontology only for the case of the merging, where it appears as the design of a new ontology operated by knowledge engineers who care for the safety of the intended meaning of the original ontologies with respect to that of the new one. On the contrary, the ontologies that we claim to be produced as the output of the other operations and processes on ontologies (other than merge) may be viewed as "natural" results, purely logical matter. Such a naturality will be more apparent in the next chapter where everything will dress the new logical vest of *Compatibility Spaces*.

For the time being we just want to sum up some crucial points that have emerged from our quick survey on ontology "operations". First of all, we find that we can restrain to far less basic "operations" in order to account for everything that can be done on and with ontologies. Namely: inheritance (and mapping) on the one hand, and union on the other. All the other operations, besides special techniques and algorithms, needed e.g. to discover in different ontologies the overlapping parts suitable for mappings, can be reduced to the ones that we have just recalled. And secondly, we highlight the uncovering of a logical rule beneath the basic, foundamental mechanism of inheritance, i.e. the *cut* rule. This will be central in our description and modelling of interaction between agents in the Semantic Web. Indeed, communication between logical structures (be they proofs or knowledge representations like ontologies) happens just thanks to the cut. That is, sticking to our discourse on operations on ontologies, thanks to the discovery of a "point" on which two ontologies agree and based on which can be attempted further communication, in sight for a greater exchange of information.

As regards the discovery of the contact points that allow to couple two ontologies, logic cannot say too much, for it depends mostly on extra-logical aspects. It requires indeed to know what the ontologies are about, what the formalized language is to represent with respect to "real", external world. Nevertheless, in the following of our work we could say something more, from a logical point of view, about how communication may set in between two ontologies (and more generally between web agents) thanks to Ludics, a theory about the logic of logical rules ([Girard, 2001]) which proposes some interesting insights on the borderline between logic and extra-logical world.

4.4. Use of folksonomies

Whereas ontologies have been the object of intense research, among different aspects, as regards the possibility of integration, comparison and exchange of knowledge based on them, folksonomies have never been considered in the setting of an inter-application exchange. On the contrary, most of the efforts on studying folksonomies focus on the effectiveness of a single folksonomy as a search utility within a specific application (that is its social environment, generally a document repository), sometimes in comparison with a "competing" ontology-based classification, or with other techniques for information retrieval, in order to assess its effectiveness. It is quite well clear that the interest on folksonomies as just local conceptual structures is rooted in the specific community environment that it comes from, since a folksonomy is exactly the sum of the many personimies resulting from the tagging activity of each community member. Then, the implicit conceptualization behind a folksonomy is relevant, in the general case, just to the small universe of the users and tagged objects of a particular web application. Nevertheless ontologies too are deeply marked by the community of experts that produce them – the pure objectivity of knowledge being a far from reach ideal – and are meaningful as far as they are used over data like those for which an ontology has been designed. Therefore, it is reasonable to ask for a knowledge exchange between ontologies and folksonomies.

Nevertheless, at present, the typical relationship between folksonomies and ontologies, is that of reduction. Indeed, before attaching value to a folksonomy as a knowledge structure, it is assumed that this structure has to be crystallized as an ontology, reduced to the form (and formal correctness) of ontology. This is precisely because of the lack in formality of the logical structure that underpins a folksonomy, that which makes hard to envisage a neat, direct comparison with an ontology.

The reduction process, called *ontology extraction*, usually relies on data mining techniques. In particular, there is a mathematical theory that allows to "mine" within a folksonomy in a very interesting manner, one which also requires little human supervision. The mathematical theory, that is the part interesting to us, is Formal Concept Analysis (FCA) ([Ganter et al., 2005]). It considers mathematical objects (lattices) corresponding to concepts, which in turn are built as sets of attributes (the tags) and groups of objects (the resources bearing tags) at the same time, thus showing the same double nature of ontological concepts (intensional and extensional aspect).

We will say something more about FCA in just a few lines, but for the time being we insist on the fact that, according to the common approach, any possible interest in using folksonomies to operate a comparison or knowledge exchange between information repositories must pass through the previous transformation in ontologies (or something like that, with a clear formalized – static – *meaning*), so that operations (processes) on folksonomies – like those we considered for ontologies – do not actually exist in the literature nor in the practice of World Wide Web.

We complain a little about that. The reason is that the reduction to ontologies causes the loss of the dynamicity of the original social environment. Indeed it implies the choice of a well definite and precise semantic interpretation of each attribute or attribute set, whereas the actual use of tags in a social tagging application always preserves the freedom to widen (or tighten) the intended meaning carried by a tag or tag association. We think that is desireable a lighter, even though weaker, logical structure apt to preserve such flexibility proper to folksonomies – and we think that at least on this point our proposal gets some score.

Although we cannot say so much about operations on folksonomies, we will spend some words on the basic logical structure of a folksonomy, a few words about FCA, and still fewer on derivated techniques, just to make appreciable a subsequent comparison with our proposal for the logical modelling of information over the Web.

4.4.1. Folksonomy: the sum of many personomies

The typical mathematical presentation of a folksonomy [Kang et al., 2009] starts talking about a tuple $\mathcal{F} = \langle \mathsf{U}, \mathsf{T}, \mathsf{R}, \mathsf{A} \rangle$ where $\mathsf{U}, \mathsf{T}, \mathsf{R}$ are sets of, respectively, users, tags and resources and A is a relation on them. That is to say, a folksonomy is a collection of attributions (A) of tags (T) to resources (R) by some user (U). Therefore, the proper objects of a folksonomy (the tag assignments) are the elements of the ternary relation $\mathsf{A} \subset (\mathsf{U} \times \mathsf{T} \times \mathsf{R})$, that is triples like $\langle x_{\mathsf{U}}, y_{\mathsf{T}}, n_{\mathsf{R}} \rangle$, which stands for "user x has annotated resource n with the tag y". As we said in section 1.3.3, a folksonomy does not deal precisely with concepts and their instances as an ontology does. Rather it collects resources under a common umbrella-term. Nevertheless such a term is not chosen by chance: it is significant according to the user who adopts it to mark those resources because all they share some common aspect, quality, attribute that precisely the chosen term denotes – or indicates, recalls, suggests. In fact, this is the lack of accuracy typical of folksonomies: it is not clear which is the relation that holds between a tagged object and the term used to tag it. Within an ontology it is typically the relation of type-specification: the name of the concept under which an individual is counted is (at least should be) the type of that individual – for some ontologists it could be its essence. Or, from a more technical point of view the concept name could be the *view* on (represented) reality from which that individual appears. Moreover, an ontology allows also to relate an individual with other concepts (or individuals that instantiate a concept) according to other relationships, such as part-of (which really attains to mereology) or "personalised" relationships like boss-of for instance. And ontologies can even relate an individual to qualities, then involving judgements and/or measurements. All this is possible thanks to the ability of ontologies to specify what every relationship stands for. But by the way, we note that even though the basic relationship of type-specification (the is-a relation) is manipulated properly by machines, computer programs and in particular by inference engines, any other "personalised" relationship is handled properly only by the application that is designed to use the particular ontology in which it appears. That is just to contain the difference between folksonomies and ontologies, which is not really too large.

Then, with a folksonomy we have terms to pick up resources out of ... in fact there is no special name for the repository of the data that a folksonomy speaks about – whereas for ontologies we speak of A-boxes to collect data (individuals and their facts), which cause the ontology to become a knowledge base when they are there. So, let's try to make more precise our vocabulary about folksonomies. We will call simply *repository* the store that contains all the resources on which users of the community have placed their tags. Now, to sum up and then fix a good mathematical (logical) presentation of folksonomies, we have to consider the three sets $\mathsf{U}, \mathsf{T}, \mathsf{R}$ of the tuple $\mathcal{F}$. The resources that enter the repository R are just the resources tagged by some user, as well as the tags to be counted in the "language" T of the folkosnomy $\mathcal{F}$ are just the tags that some user has used on at least one resource of the repository, that is to say the ones that appear in at least one assertion.

We must still deepen in more detail so as to reach the level of *personomies*. Indeed, if we remember the original definition of folksonomy by Vander Wal, that says "folksonomy is the result of personal free tagging of information and objects (anything with a URL) for one's own retrieval" the personal dimension of the tagging activity is quite apparent. Although Vander Wal does not talk about personomies, these last can be easily understood as being the "slices" of which a folksonomy is made: every personomy $\mathcal{P}_x$ is that part of $\mathcal{F}$ containing only and all the attributions $\langle \mathsf{U} \times \mathsf{T} \times \mathsf{R} \rangle$ for a given $x \in \mathsf{U}$. To say it in another, poorer way, a personomy $\mathcal{P}_{x \in \mathsf{U}}$ is a binary relation on $\mathsf{T}_x \times \mathsf{R}_x$, a formulation from which the users' dimension may disappear, provided that we restrain the vocabulary T and the repository R to only and all the tags used and the resources tagged by user x. We may note now that the users' dimension, the social component is missing in the world of ontologies. This will be the reason why we shall compare ontologies to personomies in the next chapters, for the role of actors producing the tagging (or annotation, or classification) of resources is not really absent in ontologies, rather it is hidden behind the mask of experts' objectivity, that is sold (and bought) as simply the truth about the world. Anyway, we will have space in the last part of this work to discuss longer on such a position.

For the time being, the display of personomies lets us take into account another characterizing element of folksonomies, which is considered in more detailed versions of the formal definition of a folksonomy, such as in [Jaschke et al., 2008, Pan et al., 2009]. The folksonomy's tuple then appears thus: $\mathcal{F} = \langle \mathsf{U}, \mathsf{T}, \mathsf{R}, \mathsf{A}, \Theta \rangle$. Indeed, those authors consider also the set Θ of relations among tags, produced based on the use of tags on the part of users. For instance, $\prec \subseteq \mathsf{U} \times \mathsf{T} \times \mathsf{T}$ is an order like the ones typically considered in *thesauri*, that represent orderings among terms, based on relations like `broader` and `narrower`. E.g., $t^1_x \prec t^2_x$ to say that the tag t^1 denotes (or suggests) a concept narrower than the one denoted by tag t^2 according to their use on the part of user x.

Any of such relations is relevant only to the use of a subset of tags of T on the parts of a specific user x, thus producing possibly a large proliferation of parallel orderings of tags within a folksonomy. It is also apparent that this is a trick to recover, within personomies (and then folksonomies), part of the hierachical structure that is proper to ontologies. We do not say whether it is good or bad practice, we limit ourselves to note i) that to have such orderings (or more generally relations *tout court*) between tags needs that the users explicitly dispose the tags in the right position within the relationships – an activity that is not contemplated in the typical tagging behaviour – and ii) the possibility to establish such relations depends on the particular Web2.0 application/service (or social community) that is considered, since it is a facility generally not available in most of tagging systems. A latent form of structuring tags in hierarchies, for instance, is supported in *Delicious* under the name of *tag bundles*. With a tag bundle a user can group a set of her tags under another tag. When using this last tag term, she will be able to retrieve in a quicker way the whole set of resources that bear at least one of the bundled tags – resources which are now supposed to share some common aspects beside what is expressed by the more specific tags directly stuck to them.

Anyway, in the following we will not consider such an ordering. It is primary in order to keep the broadest generality for our discourse, and then also for sake of simplicity. And by the way ordering between tags will be handled with no difficulty with our *compatibility spaces.*

For the record, there are yet other aspects that can be taken into account while giving a formal dress to folksonomies, like [Hotho et al., 2006] for instance do with respect to the temporal dimension. In this case a tag assignment carries another information concerning the moment m when user x has annotated resource n with the tag y. But we are not going to take into account such an aspect which by the way proves to be very useful for improvement of search effectiveness within tagging sysetm. Nevertheless, this kind of interest is completely out of ours own since we aim to a general formal, logical definition of information repositories over the Web, such that we can try to represent all them through our very general purpose model, with no comparison, at least for the time being, with other models as regards perfomance on query answering. Our aim, we recall, is to propose something for the, still missing, overall logical account of Semantic Web and interaction in it.

We propose then to define folksonomies in an alternative way, one that by the way will bring us one step closer to our idea of logical representation of information over the Web. Thus, a folksonomy is the sum of all the personomies produced by the users of a social community, a Web2.0 application and the like. We mean the sum in logical sense:

$$\begin{aligned} \mathcal{P}_{x \in \mathsf{U}} &= \langle \mathsf{T}, \mathsf{R}, \mathsf{A}_{x \in \mathsf{U}} \rangle \\ \mathsf{A}_{x \in \mathsf{U}} &\subseteq \mathsf{T} \times \mathsf{R} \\ \mathcal{F}_{\mathsf{U}} &= \biguplus_{\mathsf{U}} \mathcal{P}_{x \in \mathsf{U}} \end{aligned}$$

In this way one can easily think primarily of a personomy as the collection of resources known by a user $x \in \mathsf{U}$, grouped by the tags in T that the same user x does use – i.e. all that is contained in the relation $\mathsf{A}_{x \in \mathsf{U}}$, or the assignments of our Mr. x. Secondly it is more intuitive to recognize a folksonomy as the result of taking all together those personomies. Indeed, also from the notation used it is apparent that a folksonomy is determined by the community of users (the set U) that operates in the specific social environment that stores the resources on which they stick their tags.

The most interesting aspect in all this is that any part of a folksonomy is open and there is a strict dependence between parts. The set U of users is just the list of registered accounts to the community, it is the most stable part, though new accounts can be activated at any time, and others could be deactivated for people leaving the community. The collection R of resources known by the community is constantly increasing as users tag new stuff (be that photos, scientific papers, books, web-pages, videos and so on). Finally the "language" (maybe vocabulary sounds more proper) of the folksonomy, that is the set T of tags, is greatly unstable not only for the continuous addition of new entries, but also for no tag has a well defined meaning and its "denotation" (the set of resources that it recalls) also changes continously (normally increasing), thus involving shifts of the possible corresponding concept at the intensional level.

4.4.2. Concepts in folksonomies

What is precisely a concept in a folksonomy is not too easy to say. It is quite easier to say what it is within a personomy instead. First of all, from a strictly technical point of view, it is really similar to that which is a concept in a Semantic Web ontology: a term that allows to take together a set of objects. Nevertheless a concept in an ontology takes also a richer connotation deriving from the philosophical tradition of ontology, which is not into play in folksonomies. In a personomy then one may call a concept what is behind the term that is used to tag a set of resources. Such a term is usually a word in some natural language, but it is not mandatory – apart from more sophisticated constructs like polirematic words, one may consider as quite common cases acronyms, abbreviations and the like, all subject to the great liberty of the users who create tags. And even when the tag term is a simple natural language word, that does not provide us with a more certain *semantic* of the tag, quite the contrary instead! Like in real life, indeed, the meaning of a word in use is something more complex than the definition provided in a good vocabulary, so also in a personomy it is not easy (in principle it could be not even possible) to guess which particular aspect of the concept that is recalled by the tag term the user was willing to highlight. Far more than the choice of terms to denote concepts in a formal ontology, is the internal structure defined by means of a logical formalism that which make clear what is the intended meaning. And the lack of such an internal structure in folksonomies is yet more apparent in the single personomy.

So, basically a tag term stands to mark some property (in the broadest sense) that applies to each of the resources that bear that tag. The way of generating concepts, one sees, is exactly the opposite to that of ontologies. In ontologies a concept is stated and formally defined, and then resources, individuals that are instances of that concept, can be found and recognized in the world onto which the ontology is posed. Thus, the objects take the form (and with that we mean in particular all the logically defined properties, also entering in the "inheritance line of succession") of the concepts of which they are recognized as instances. In a personomy, on the contrary, one abstracts a property based on the resource he is dealing with; and the more resources one tags with the same tag, the more the concept behind the tag gets altered, enlarged in order to accomodate the increasing quantity of resources – as well as one passes from the essence of a specific individual to the common characterizing property of more and more similar yet distinct individuals. So, we can conclude that a concept in a personomy is the undefined abstraction of a (quite elastic) property that applies to all the resources that a user annotates using that tag.

We can get some help in defining what is a concept in a personomy with the support of Formal Concept Analysis [Jaschke et al., 2008, Kang et al., 2009]. FCA, as well as formal ontologies for information systems, relies on a neat distinction of two components of the concept of "concept", that is its intension and extension.

Now we briefly present the basic ideas of FCA, readjusting the specific terminology in order to immediately fit the discourse on folksonomies. A key element of FCA is the notion of formal context. It is actually, with respect to folksonomies, nothing else than what we called personomy. Indeed, it is defined as a triple counting a set of objects, a set of attributes and a relation on them. If we call the objects resources (R), the attributes tags (T), and their relation assignment (A) a formal context will look like $\langle \mathsf{T}, \mathsf{R}, \mathsf{A} \rangle$, with $\mathsf{A} \subseteq \mathsf{T} \times \mathsf{R}$.

FCA goes then a little further in characterizing formal concepts and provides this defintion:

Definition 1 *Given* $\mathsf{T}' \subset \mathsf{T}$ *and* $\mathsf{R}' \subset \mathsf{R}$*, the pair* $\langle \mathsf{T}', \mathsf{R}' \rangle$ *is a formal concept if and only if* ${\mathsf{T}'}^{\mathsf{A}} = \mathsf{R}'$ *and* ${\mathsf{R}'}^{\mathsf{A}} = \mathsf{T}'$*, with:*

$$\begin{aligned} {\mathsf{T}'}^{\mathsf{A}} &= \{ r \in \mathsf{R} \,|\, \forall t \in \mathsf{T}' \quad \langle r, t \rangle \in \mathsf{A} \} \\ {\mathsf{R}'}^{\mathsf{A}} &= \{ t \in \mathsf{T} \,|\, \forall r \in \mathsf{R}' \quad \langle r, t \rangle \in \mathsf{A} \} \end{aligned}$$

That is a formal concept is the reciprocal closure between resources tagged with the tags in T' and tags used for the resources in R'. It is easy to see now in which sense T' is called the *intension* of the formal concept and R' its *extension*.

It is noteworthy that the definition of formal concept takes into account the more general case of a set of attributes (tags) to form one concept – more general than our approximation of concepts in personomies, which for sake of simplicity was presented just referring to single tags. From now on it is to be considered always the FCA version of intension of a (formal) concept, of which a special case is a singleton of attributes (the single tag). Note also, by the way, that normality in tagging environments, where personomies and folksonomies come from, is precisely the case of multiple tags on a resource.

Now that is clear what is a concept in a personomy, things will get quite confused again as soon as we scale up to the whole folksonomy. Indeed, due to the multiplying factor given by the set U of users, we just should have a huge proliferation of concepts unrelated to each other, because even homographic tags used by different users may lead to different concepts.

Yet precisely at this point the social component of a folksonomy turns out to be its real treasure. It allows to discover a peculiar structure within the folksonomy. A structure that frames users' concepts in a larger picture, based on the observation of community dynamics. Indeed by observing the social use of tags one can measure similarity of use of tags on the part of some users, as well as their interest for the same resources thus suggesting the presence of common ideas behind certain tags and detecting the existence of sub-communities. As many researches put it out (cf. [Mika, 2007] just to name one), even though a single personomy is just an idiosyncratic catalogation, the social community dimension allows to recover something quite reliable to be accepted as the emerging semantic of a folksonomy ([Mika, 2007]). On the contrary, ontologies completely miss this aspect, even though as "explicit specifications of shared conceptualizations" [Gruber, 1995] they are the result of the agreement of some (likely pretty small) community about which aspects of a domain are to be considered. That is not so different with respect to what people do in Web2.0 communities, where objectives like suggesting interesting web-pages or showing nice pictures make users co-operate, collaborate in re-using most effective tags. As a consequence, quite rich and articulated structures of tags emerge. Ontology extraction techniques exploit precisely these mechanisms. But once the ontology is extracted, we are out of the dynamical social environment. This is the reason why we would rely on a less formalized model than ontology, yet logically meaningful.

From a more concrete point of view, the discovery of the internal social and shared conceptual structure is achieved by means of a variety of algorithms rooted in the principles of FCA. Each one looks at different "footprints" of social interaction and operates by measuring different selections of parameters. Some, for instance, consider other mathematical objects built on top of FCA, like *triconcepts* ([Jaschke et al., 2008]), which allow to recover the "tridimensional" reality of folksonomies (users $\times$ tags $\times$ resources). Others ([Kang et al., 2009]) rely directly on FCA and deploy algorithms to iteratively evaluate clusters of "formal concepts" built time by time out of pairs users $\times$ tags, users $\times$ resources and resources $\times$ tags. One need just consider that even the basic techniques of data mining known as basket analysis (that which enables many e-commerce websites to propose us the items that other customers have bought together with the item we are going to buy) exploit the same principles of FCA.

Sure, that is all statistical analysis. It allows to discover and highlight a network of users that share interests on some objects and classify them with same words. Then such algorithms point out a small community, or a shared concept according to what they are set to detect. But the expected following step is evaluation of results by human, and fine tuning of the so extracted ontology. Thus we are again out of our field of interest. We are not interested on these ways of extracting "potentially regular" concepts; we prefer to stay at the previous step of analysis, that is up to personomies and the fine definition of concepts that FCA provides us with. And then we will try to model interaction between personomies (and whatever else form of information storing over the Web) in an alternative way.

4.5. Ontologies vs folksonomies

Beyond the number of specific implementations for the processes of use of ontologies (what we have also generally called operations on ontologies), they all can be grouped and recognized as tasks that act primarily and essentially on the conceptualizations that are behind the ontologies, requiring the skills of human experts to understand the knowledge that is encoded there (the description provided for some, more or less specific, domain) in order to perform an activity that looks more like a philosophical task of comparison, evaluation and adjustment of culturally and linguistically (and even more) biased mental schemata, rather than a matter for automatic reasoning and interconnected machines.

In fact, as we said in the previous part of this work, we are interested in methods and techniques that allow for computer programs (web agents) to share knowledge as freely as possible, that is in particular with no need for given in advance declarative specifications that enable the exchange of data between agents and datasources. This is also the reason why we look at A-boxes and take care of the extensional level of ontologies: this is the low-level information that machines can manage, so that we would like to track a way out for web agents to play an even very poor, but at least somehow significant interaction and sharing of data in a meaningful fashion.

It is apparent that these two dimensions are incomparable, that of conceptual knowledge with respect to that of punctual information, both stored in ontologies. The former is accurately prepared in order to give structure to the latter; this is also the reason why the interest is focused on the structure, not on the content. So, we do not even think that the part that we are "supporting" deserves *more* attention than the other. Anyway we claim that it deserves *some* attention, whereas it seems to have never got sufficient interest.

If we abstract from the high-level conceptual structures and we focus on simple data, we can find basic directions about what a piece of information is. We may not need to know everything that could be said about the concept that is instantiated by that particular piece of information – by the way not even the richest and largest ontology could say everything. Nevertheless we could be happy with this: just a sign, a tag, which marks our information object and makes it comparable with others. It is what actually a machine (a computer program, a web agent) can grasp. Then we can compare any information object to many other objects and possibly find some others that bear the same sign, that is objects similar to some extent. The machine would not *really* know what that object is neither if the object were annotated according to the largest existing ontology. Then, basically it is just a matter of richness of the additional information, the metadata (the "signs" on which to perform comparison), and of trust that we attribute to who placed the tag, added the metadata.

Indeed, with an ontology and data annotated according to that ontology – and the same holds with data *viewed* through an ontology interface, *via* queries over a database or other repository – we have also the guarantee that information has been properly registered, that the annotation is correct i.e. sounding with respect to the real world and the actual state of things in the world[14], with one word we may *trust* that information. This is the reason why ontologies are adopted within large and important organization to organize and manage quite sensible, valuable data. On the contrary there is a sort of "trust issue" with the Social Web or Web2.0. What is the reliability of socially categorized information? And up to which kind of information or data would we rely on social tagging to get our information about? Really there are some successful experiences of social web, among the first and best known ones we name just Wikipedia, which have gained a lot of trust both from web users and service providers.Nevertheless, many ontologies deal with very specific domains like e.g. genes and their relations with (genetic) diseases and no one would even think that the same domains could be tackled on the basis of socially structured knowledge systems. Therefore it is apparent that the appropriateness of an approach or the other depends mainly on the particular subject that an application or service is intended to deal with, according to the level of trust that is required to perform any activity involving information on that subject.

Thus, once it is clear that trust is beyond our focus in this work[15], we would con-

[14] At least according to the point of view of the "ontologists" who have designed and mantain the ontology – it is the same implicit assumption one must adopt even when reading on Wikipedia.

[15] As said in the first part, with respect to the figure 1.1 we focus our research and interest on the central area, involving ontologies among other kinds of structures for information over the web

centrate on how roughly, poorly annotated information could be directly used to have communication between computer programs (agents) happen on a large scale and dynamically over the Web.

We propose then a logical framework where information from ontologies, folksonomies and possibly from any kind of information repository connected in the Web can be put on the same, unique basis and processed according to some simple logical operations. Moreover, once given this new form to information we can ground on top of that a general model of interaction between web agents. In the following we present these proposals. In the margin, we remark the deep difference that is between such an idea and other, yet interesting, proposals that aim to reduce the distance between ontologies and folksonomies, and with those also between Semantic Web and social web, by regimenting free tagging within the norms of neat, well hierarchised ontologies, so as to enable its use with inferencial engines or at least to bring it back to the "style" of the mainstream Semantic Web ([Gruber, 2007]).

and the overall logic that should make everything communicate – in a sense higher than Internet Protocol.

5. Ontological Compatibility Spaces

We propose here a theoretical framework for knowledge representation, and more generally for information exchange, specially conceived for use within the Semantic Web scenario. Instead of focusing on conceptual schemas of ontologies, we propose to focus on the extensional level of knowledge bases, i.e. on "real" objects, by adopting a logical framework capable to geometrically represent relations among resources, possibly discovering the concepts from resource aggregations that are actually in the Web. In particular we suggest to consider another kind of logical interpretation that relies on structures richer than sets, the *compatibility spaces*, where the interpretation of a concept (or tag-term) produces graph theoretical objects along with the determination of the extensional counterpart within the collection of resources that is the domain of interpretation.

5.1. One Step Further with More Structure

Compatibility spaces are actually the same as Coherence spaces [Girard et al., 1989], i.e. are webs whose points may or may not be linked to each other according to a binary symmetric reflexive relation called *coherence*. We call them here *Compatibility spaces* (and *compatibility* their structuring relation) so as to distinguish our peculiar use as theoretical objects to represent the information sparse in the World Wide Web. And also to be appropriate: we cannot claim about coherence of Web resources, but we can show their compatibility under certain conditions, with the aim to address Semantic Web interest in the integration of independent data sources and, more generally, in knowledge and information sharing.

Coherence spaces allow for the definition of a denotational semantic, so that, through Compatibility spaces, we can get one also for data exchange within the Web and for a definition of operations between ontologies that is primarily focused on resources. Coherence spaces come from Linear Logic (LL) and have been the first semantic interpretation of that logical system. What is more is that it is not truth-valued semantics, useful only for talking about formulas – as it always happens with respect to Classical Logic (LK) – but precisely denotational semantics, useful for talking also about proofs. Indeed, they offer the domain of interpretation of the objects manipulated by the logical calculus, i.e. proofs.

LL [Girard, 1987] is a logical system developed by imposing some restrictions on the use of structural rules for the construction of deductions within Gentzen's proof calculus for First Order Logic known as sequent calculus. The affected rules are Contraction and Weakening which deal with the number of times formulas may be used within the same proof. In LL they are re-defined in form of logical rules (instead of structural) so that their usage has to be marked with specific connectives, called *exponentials*. This way everything that holds in LK also holds in LL, although LL is able to better describe what is happening in a proof: the ability to mark for which formulas it is licit to have weakening and contraction means that one can control the times resources are used. We note, by the way, that this one looks like an interesting property to have at hand while working for a Semantic Web of resources. As a consequence of the control on contraction and weakening, LL deconstructs the connectives $\wedge$ and $\vee$ doubling them in the multiplicative and additive variants, since their behaviour is different according to the possible uses of the context (i.e. the other formulas) where the formulas in which they occur are interacting with. To put it in a nutshell, the multiplicative connectives operate on the coherence space resulting as the product of the coherence spaces corresponding to the proofs of the connected formulas, while the additives on their disjoint union.

Moreover, LL has developed a geometrical representation of proofs by means of graphs called proof-nets. It exploits graph structures to compose partial proofs and provides graph properties to determine when a proof structure is correct. Such graph structures have a model in coherence spaces, where the denotation of the proof of a formula is a set of pairwise coherent points, called *clique*. The operations between coherence spaces interpret the composition of formulas and their proofs according to LL connectives.

It is time to formally introduce Coherence spaces, so that we will be able to show also by comparison the peculiarities of our Compatibility spaces. We recover their definition from [Girard, 2006].

Definition 2 (Coherence spaces) *A coherence space X is defined by its:*

support *: the underlying set of points, noted $|X|$*

coherence *: a binary, reflexive and symmetric relation between points of $|X|$, noted $x \frown_X y$*

A subset a of $|X|$ whose points are all pairwise coherent is called clique*, and is noted $a \sqsubset X$.*

An alternative definition of Coherence spaces, equivalent to the previous one, was given in [Girard et al., 1989] and is based on the notion of clique:

Definition 3 *A coherence space is a set (of sets) X that satisfies the two conditions of* down-closure

- *if $a \sqsubset X$ and $a' \subset a$, then $a' \sqsubset X$*

and binary completeness

- *if $Y \subset X$ and $\forall a_1, a_2 \in Y\ (a_1 \cup a_2 \sqsubset X)$ then $\bigcup Y \sqsubset X$.*

In order to produce a meaningful interpretation of, especially, ontologies by means of coherence spaces we need a particular class of them with a typical support set (see Definition 5). As a consequence, we propose also to call this special class of coherence spaces "Ontological Compatibility Spaces" (OCS).

5.2. Ontological Compatibility Spaces

Let $\mathcal{O}$ be a semantic web ontology in a language like OWL and $S(\mathcal{O}) = \langle Voc(\mathcal{O}), Ind(\mathcal{O}) \rangle$ its signature. $Voc(\mathcal{O})$ is the set of all the symbols for concepts and relations[1] of $\mathcal{O}$ (i.e. unary and binary predicate symbols), whereas $Ind(\mathcal{O})$, possibly empty, is the set of individual names (i.e. individual constants)[2] appearing in $\mathcal{O}$.

As we said in the previous chapter, if $Ind(\mathcal{O})$ is empty then we call $\mathcal{O}$ just a terminology – a vocabulary with logical definitions for the terms such that it expresses a conceptualization of a domain, with no commitment to any specific state of affairs. If $Ind(\mathcal{O})$ is not empty, we call $\mathcal{O}$ a knowledge base (KB), for it specifies some state of affairs in the domain that it represents by stating *facts* about individuals – in this case we have an A-box containing assertions about individuals along with the T-box containing the terminology. Actually, this is true if one considers ontologies form the perspective of Knowledge Representation, from where the notions of T-box, A-box and KB itself derive. It is perfectly correct also with Semantic Web ontologies, provided that they are written in a language implementing some DL, that is five out of six dialects of OWL. On the other hand, the discourse is slightly different for the ontologies that are produced, with the OWL-Full dialect or RDF, in that other line of development of Semantic Web that simply does not care for DLs and inferential engines, and strictly speaking does not produce KBs. Nevertheless, wherever such a line contemplates just RDF schemas (as well as OWL-Full ontologies), wherein vocabularies are defined, and a number of files, sparse over the Web, which are compliant with those schemas, we can reasonably see, respectively, detached T-boxes and A-boxes, that together form, they too, knowledge bases.

Hence we introduce another way to produce a KB even when the ontology $\mathcal{O}$ is not DL-compliant, or it is another form of knowledge representation / information storage in the Web. Given a collection of resources[3] M, it forms a KB together with $\mathcal{O}$ iff there is a valuation ϕ from $Voc(\mathcal{O})$ to $\mathfrak{P}(M) \cup \mathfrak{P}(M \times M)$ such that for every predicate symbol $P \in Voc(\mathcal{O})$, $\phi(P)$ retrieves the set of resources in M that satisfy predicate P (possibly through a special mapping, for instance when M is a database

[1] Relations are usually called properties in OWL, with the distinction between datatype properties and object properties. The former ones are actually attributes, the latter roles (cf. section 1.2.5). Apart from OWL and the Semantic Web languages, also some DL languages offer the possibility to deal with attributes. Without loss of generality however, within the frame of our discourse about Ontological Compatibility spaces we will deal with both in the same way, as if they all would be roles.

[2] Like for relations, we should distinguish between strictly speaking individual names – that is terms that denote some instance of a given concept – and data values. Data values are the last ends of triples (in the typical RDF, but also DL, style of expression subject-predicate-object) involving attributes. Whereas the subjects are always true individuals, if the predicate is an attribute then the object must be a data value. Nevertheless, data values are instances of some datatype and the really ontological issue of determining the possible difference between concepts and datatypes gets shadowed in formal languages for ontology by much more technical and pragmatical considerations. As a consequence, as we disregard attributes, so we disregard also their data values, since these last are homogeneous to individual names from the viewpoint of a formal language like the one that we are defining.

[3] In a very general sense: it can be an RDF repository, an XML formatted document or even a database, in which case a resource is a row in a table (a *record*) according to the relational data model.

and ϕ fires queries against its logical schema to retrieve sets of data).

Definition 4 (Knowledge base) *Formally, for P and R respectively unary and binary predicate symbols in $Voc(\mathcal{O})$,*

$$\begin{aligned} \langle \mathcal{O}, M, \phi \rangle \textit{ is a KB } \Leftrightarrow \quad & \forall P \quad \phi(P) \subseteq M \\ \textit{and} \quad & \forall R \quad \phi(R) \subseteq (M \times M) \end{aligned}$$

We take this last as the general case, since even in the case where we have an A-box for $\mathcal{O}$ we can easily take $Ind(\mathcal{O}) = M$ and there is a "concrete" function ϕ that recovers the subsets of M that satisfy any predicate from $Voc(\mathcal{O})$: it is the (automatic) reasoner – an inference engine – that normally empowers a KB made of T- and A-box together.

The triple $\langle \mathcal{O}, M, \phi \rangle$ defines a knowledge base[4] and we can represent every concept and relation of it by means of a special kind of coherence spaces that we call ontological compatibility spaces.

Definition 5 (Ontological Compatibility Spaces, OCSs) *Based on a KB $\langle \mathcal{O}, M, \phi \rangle$, an OCS $[\mathcal{O}, M, \phi]$ is defined as follows:*

- *its support[5] is $|[\mathcal{O}, M, \phi]| =$*

  $= \{x \mid x \in M \wedge \exists P \text{ s.t. } x \in \phi(P)\} \cup \{\langle x, y \rangle \mid \langle x, y \rangle \in M \times M \wedge \exists R \text{ s.t. } \langle x, y \rangle \in \phi(R)\}$

  for P and R respectively unary and binary predicate symbols in $Voc(\mathcal{O})$

- *the compatibility relation between the points of $[\mathcal{O}, M, \phi]$, noted $\frown_{[\mathcal{O},M,\phi]}$,[6] is assigned according to the following:*
  - *for $x, y \in M$*

    $$x \frown_{[\mathcal{O},M,\phi]} y$$

    if and only if there is a unary predicate symbol P of $Voc(\mathcal{O})$ such that

    $$\{x, y\} \subseteq \phi(P)$$

  - *for $\langle x, z \rangle, \langle y, w \rangle \in M \times M$*

    $$\langle x, z \rangle \frown_{[\mathcal{O},M,\phi]} \langle y, w \rangle$$

    if and only if there is a binary predicate symbol R of $Voc(\mathcal{O})$ such that

    $$\{\langle x, z \rangle, \langle y, w \rangle\} \subseteq \phi(R)$$

[4] In [Abrusci et al., 2009, 2011] we called it *applied ontology* in order to reflect our interest with ontologies that allow to share punctual information in the special context of the World Wide Web by highlighting the aspect of shareability, better suggested by the word ontology, rather than that of fine classification of data, clearly recalled by the term knowledge base. Nevertheless, we recover here the traditional term in the field of knowledge representation in order to avoid misunderstandings.

[5] Usually the support of a coherence space X would just be noted $|X|$; however we are dealing here with collections of resources described by semantic web ontologies. These last can have only unary or binary predicates in their language. Based on the notion of compatibility, that will be yet clearer in next lines, it is useful to "reify" as objects of a compatibility space also the pairs that fit into some relation specified in an ontology, along with single individuals.

[6] For sake of clarity we may note simply $\frown$ when the compatibility space is obvious.

– *and also, for* $\langle x,z\rangle \in |[\mathcal{O},M,\phi]|$

$$x \frown_{[\mathcal{O},M,\phi]} \langle x,z\rangle,\ z \frown_{[\mathcal{O},M,\phi]} \langle x,z\rangle \text{ and } \langle x,z\rangle \frown_{[\mathcal{O},M,\phi]} x,\ \langle x,z\rangle \frown_{[\mathcal{O},M,\phi]} z$$

Definition 6 (Clique) *As for general coherence spaces, a group a of pairwise compatible points of $[\mathcal{O},M,\phi]$ is called a clique, and is noted $a \sqsubset [\mathcal{O},M,\phi]$. More formally:*

$$a \sqsubset [\mathcal{O},M,\phi] \Leftrightarrow a \subset |[\mathcal{O},M,\phi]| \wedge \forall(x,y) \in a,\ x \frown y$$

In particular, $\emptyset \sqsubset [\mathcal{O},M,\phi]$.

We observe that the compatibility relation formalizes the notion of compatibility emerging whenever an ontology is applied to a set of data. Indeed, from an abstract point of view, the retrieved values for any predicate symbol in $Voc(\mathcal{O})$ form some subset of $M \cup (M \times M)$ whose elements share with each other something more than with any other element of $M \cup (M \times M)$. Such a property, instead of being named according to any specific predicate symbol P occurring in the ontology, may be rewarded as the compatibility between all the objects of that subset of M.

Ontological Compatibility Spaces can be considered as models for knowledge bases. However that would not be so interesting since there is already a well established and rich literature concerning standard interpretations by means of Model Theory and, perhaps less standard, graph theoretical structures, even for the case where KBs are publicly exposed in the Web (cf. for instance the "RDF Semantics" specification by the W3C [Hayes, 2004]). Rather, we believe that OCSs deserve interest and attention for the possibility that they offer to observe and study from a *geometrical* point of view what is, and what should be, interaction among ontology-aware-agents that act on resources in the (Semantic) Web. We will come back on this point later in this chapter (section 5.5) and in the next one.

A few comments about the properties of the compatibility relation in OCSs. Defined like the coherence relation of standard coherence spaces, the compatibility relation too is reflexive, symmetric and non-transitive.

Thanks to **reflexivity** ($\forall x \in [\mathcal{O},M,\phi]\ x \frown x$), every point of the comaptibility space is a clique and may be considered as the minimal class of compatibility.

Symmetry ($\forall x,y \in [\mathcal{O},M,\phi]\ x \frown y \leftrightarrow y \frown x$) expresses the core of the idea of compatibility, since for compatibility we mean the possibility to put two objects (resources) together based on the fact that they share some common property – not necessarily one expressed by means of a special predicate symbol – and such a commonality has to be inevitably a reciprocal fact.

Non-transitivity (i.e. it does not hold for any $x,y,z \in [\mathcal{O},M,\phi]$ that $(x \frown y \wedge y \frown z) \rightarrow x \frown z$) prevents the overwhelming distribution of compatibility that would mix up the sets of objects corresponding to different predicates of $Voc(\mathcal{O})$ whenever they share some common element. In particular, in the case of an ontology, transitivity of compatibility would make merge any two sets of individuals corresponding to distinct concepts as soon as there is at least one individual that instantiates both concepts – which is clearly not a faithful behaviour with respect to the ontology.

Remark 1 *We observe that the representation of knowledge bases as OCSs succeeds:*

- for every predicate symbol $P \in Voc(\mathcal{O})$, $\phi(P) \sqsubset [\mathcal{O}, M, \phi]$
- following the inverse direction, every clique of the OCS is the denotation:
  - of some predicate symbol of $Voc(\mathcal{O})$;
  - or of a *new* predicate not identified by any predicate symbol of $Voc(\mathcal{O})$ which corresponds either to a subconcept of some other concept specified in $Voc(\mathcal{O})$ or to a new concept "spreading" over different concepts already identified in $Voc(\mathcal{O})$;[7]
  - the empty clique $\emptyset$ corresponds to an empty, somehow "impossible" concept.

Proposition 1 *Every Ontological Compatibility space (OCS) $[\mathcal{O}, M, \phi]$, as defined above, represents a knowledge base*[8] *and is a Coherence space, therefore the properties of Coherence spaces hold for it.*

Proof. It is apparent from the definition of OCS. □

It is worth considering here what the two conditions of down-closure and binary completeness (cf. Definition 12 of Coherence spaces) mean with respect to knowledge bases (and in particular to ontologies). First of all we may verify that also OCSs satisfy them. As regards *down-closure* it is immediately shown. In particular, if we state down-closure for the OCS $[\mathcal{O}, M, \phi]$

$$a \sqsubset [\mathcal{O}, M, \phi] \ \wedge \ a' \subset a \ \rightarrow \ a' \sqsubset [\mathcal{O}, M, \phi]$$

$a' \sqsubset [\mathcal{O}, M, \phi]$ holds necessarily because, for the definition of clique, $\forall x, y \in a$, $x \supset_{[\mathcal{O},M,\phi]} y$, where the compatibility derives from some predicate symbol $P \in Voc(\mathcal{O})$ (for the definition of compatibility). As a consequence, every subset $a' \subset a$ will be made of elements all pairwise compatible, at least for the same $P \in Voc(\mathcal{O})$ that builds up a, so that they still form a clique – and we can go down to singletons, which still are cliques for the reflexivity of the compatibility relation, and the empty clique which is by definition a clique of any OCS.

Concerning *binary completeness*, it is expressed by

$$N \subset [\mathcal{O}, M, \phi] \ \wedge \ \forall a_1, a_2 \in N(a_1 \cup a_2 \sqsubset [\mathcal{O}, M, \phi]) \ \rightarrow \ \bigcup N \sqsubset [\mathcal{O}, M, \phi]$$

---

[7] This last case appears as a possible way to discover new concepts, and it looks especially interesting with respect to folksonomies.

[8] We remark that *knowledge base*, as we have defined it above, is a very general term that stands for any information repository (in particular those accessible in the World Wide Web) equipped with a logical structure (typically an ontology) that defines the objects that the repository accounts for (its resources), and which can be concretely produced by means of a variety of different technical and logical instruments, such as e.g. databases described by ontologies written in some W3C standard language, or *stricto sensu* knowledge bases made of T-box and A-box, or even folksonomies as we will see at the end of this chapter.

where N is a set of cliques. Proof is quite straightforward: given any two points $x, y \in \bigcup N$ such that $x \in a_m \in N$ and $y \in a_n \in N$ it holds that $x \frown_{[\mathcal{O},M,\phi]} y$ since, by hypothesis, every $a_m \cup a_n$ forms yet a clique and $x, y \in a_m \cup a_n$. Hence all points in $\bigcup N$ forms a clique and $\bigcup N \sqsubset [\mathcal{O}, M, \phi]$.

However, down-closure is not to be considered too strictly as if it would say that for every subclique of a clique of $[\mathcal{O}, M, \phi]$ there is a corresponding predicate symbol in $Voc(\mathcal{O})$ that provides (and justifies) the compatibility between any two points of that subclique. Indeed the definition of compatibility tells us that predicates induce cliques and not the other way around.

A predicate $P \in Voc(\mathcal{O})$ is typically considered as a concept name (or role name), and is supposed to express some concept (or relationship). Now, to precisely say what a concept is in general would be too difficult and awkward here – we will face this question in the third part of this work, where the philosophical setting of the discourse will better accommodate its (huge) dimension. For the time being we will consider an understanding of what is a concept quite limited, but relevant to the dimension of Knowledge Representation (as a technical discipline). A concept name in an ontology, then, is a sort of interface between human ideas (the concept) and recorded data (facts in the jargon of KR), precisely as attribute names in the logical schema of a database, or column headings in any table. To call it *concept name* depends on the "ambition" of ontologies to be rich, expressive and high-level (as this last expression is used for programming languages) descriptions of some well circumscribed world (the domain of interest, or universe of reference) so that it should correspond to some complex idea, conveniently simplified. Nevertheless, a concept name keeps its ability to give access from the intensional dimension of a language to the extensional dimension of occurring "reality". Even though this reality is built by means of symbolic representations, a language indeed, it *is* yet the actual reality for the complex system composed not only of the machines and software running on them that empower an informative service, but comprising also the people that access and use it, and *accept* and trust that reality as it is returned after every particular request.

Before going too far in "socio-informatics" considerations, we can more profitably observe what happens with this *access function* of concept names (our predicate symbols). The access indeed works neatly in only one direction: from the language to reality, and this holds both for the special languages built with ontologies and for any natural language. Indeed, in a language we have terms (natural words and concept names from ontologies are the same to this respect) to refer to aspects of reality that we are interested in, so that these terms act as handles to manage parts of reality. In the inverse direction we have a much less definite correspondence: if one attempts to look at reality leaving aside the language, even considering only material reality, she may find out a number of properties of things for which her language has no specific word, since basically these are not interesting aspects, of no use for normal human activities – by the way, natural language is pretty quick to coin new words for emerging needs.

Such an understanding of concepts may be compared to *types* usually considered in Logics and in computer science. Types are often said to express the meaning of other objects, as with typed calculi, for instance, where types give the meaning of a logical proof. Such a meaning is actually the expected (and foreseen) use of the resources handled during the process. It is even more apparent in software programming: types make input data identifiable – allowing to determine what the process can do with them – and the type of the output data is the warranty that the output responds to the requests sent to the program. But type systems are quite rigid systems: one cannot access aspects of the reality (the resources handled) which have not been already given a type, for simply one cannot observe reality if not through the collection of types made available – as well as one cannot expect for an unforeseen type of output to emerge after the processing of a request.

From this perspective, then, our OCSs are somehow more liberal, thanks to the possibility to discover new types, new possible uses of the resources that emerge from unexpected requests. The crucial point seems to be the "window" left open on reality – the possibility to look directly at resources and at their compatibility, from a merely extensional point of view – that allows to take new views over them, different from the ones already defined and recorded. That is much more similar to the capability of natural languages to coin new words in order to grasp aspects of reality that were deemed not interesting before – even though we may choose not to fix any term to record new views over an OCS.

Like with natural languages, with OCSs we have the asymmetric relation between language and reality: for every concept name $P \in Voc(\mathcal{O})$ we are able to recover the clique $a \sqsubseteq [\mathcal{O}, M, \phi]$ of all the resources in M (points of the OCS) that *realize* the concept expressed by P within the (very limited) universe of the data source that we are looking into (be it an A-box, a database or whatelse). *Viceversa*, we may not be able to identify a specific concept name P in $Voc(\mathcal{O})$ for every set of compatible resources that we may observe as forming a clique in $[\mathcal{O}, M, \phi]$. For we may, very likely, pick up some clique a which is just a subclique of some other clique b, with b being the *realization* of some concept name P, thus being an "interesting" clique, and a being a (sub)set of resources that nobody would have any interest in isolating from b as deserving a distinct concept name to be handled – the reader can see here the parallel with the aspects of reality that miss a word to be named since they are of no interest to anybody.

Nevertheless, we may hold that, to some extent, every clique – therefore including also any uninteresting subclique of an interesting clique of $[\mathcal{O}, M, \phi]$ – corresponds to some concept, but it may be the case of a concept for which there is no (maybe not yet) corresponding predicate symbol in $Voc(\mathcal{O})$ – and possibly not even a term in any natural language that would gather precisely that set of resources. After all this is not a "problem" or a leak of this logical model for knowledge bases: on the one hand indeed it appears to be a perfectly natural problem of reference by which any language suffers; and on the other hand we are precisely attempting to escape from linguistic (and in particular natural languages) constraints to semantics of data. Therefore we can accept to deal with unexpressed (and possibly unexpressible) concepts that we can find thanks to the mere observation of compatibility among resources. By the way, it is nothing far from the way how techniques for data-mining and ontology extraction work. And the gain with all this will be the possibility to discover new "concepts", linguistically uncovered, suggested solely by the observation of occurring reality, that is collections of resources that respond to some unforeseen purpose of use.

To conlude, for the moment, with considerations on cliques of the OCSs, the final concern is with binary completeness. The set N indeed is not forced to be neither the denotation of a full concept of $[\mathcal{O}, M, \phi]$ represented by a single predicate symbol of $P \in Voc(\mathcal{O})$, nor the denotation of a subconcept of such a concept, since, briefly, it is not necessary that any two points of N are compatible exactly for the same $P \in Voc(\mathcal{O})$ to form a clique (cf. Definition 7).

Remark 2 *There is a major shift in the interpretation of ontological compatibility spaces with respect to the standard interpretation of coherent spaces.*

As Girard puts it out in [Girard et al., 1989], the aim with coherence spaces is "to interpret a type by a coherence space X and a term of this type by a point [i.e. a clique] of X (coherent subset of $|X|$, infinite in general)"[9]. Precisely because of infinite points he needed the notion of *finite approximation*: "An approximant of $a \in X$ is any subset a' of a". Girard observes that there are always enough approximants for a:

- a is the union of its set of finite approximants;
- the set I of finite approximants is *directed*, that is
  - I is nonempty ($\emptyset \in I$)
  - if $a', a'' \in I$ there is $a \in I$ s.t. $a', a'' \subset a$, (take $a = a' \cup a''$)

In particular, Girard remarks that along with true, total objects of X there are also approximants, partial objects, of which $\emptyset$ is an example. A possibility to identify true, total objects may be in the simplest cases – according to Girard – to think of them as the maximal cliques.

Definition 7 (Maximal clique) *A maximal clique is a clique a such that*

$$\forall x \in |X| \; (\forall y \in a \; x \supset_X y) \; \Rightarrow \; x \in a$$

In other words, a clique to which cannot be added other elements unless failing the condition of pairwise coherence among all elements. We may note that within OCSs we can have two different readings of maximal clique depending on the quantifier one uses in front of the predicate that should assign the compatibility to any two points of that clique. With respect to the usual OCS $[\mathcal{O}, M, \phi]$, we said (see Definition 5) that two points $x, y \in M$ are compatible if and only if there is a predicate symbol $P \in Voc(\mathcal{O})$ such that $\{x, y\} \subseteq \phi(P)$. Now, the definition of maximal clique for Coherence spaces leaves room, in our setting of OCSs, for a broader and a narrower notion of compatibility that we may express by rewriting the definition here above in two different ways, respectively:

$$\forall x \in |[\mathcal{O}, M, \phi]| \; (\forall y \in a \; \exists P \in Voc(\mathcal{O}) \text{ s.t. } \{x, y\} \subseteq \phi(P)) \; \Rightarrow \; x \in a$$

[9] J.Y. Girard, Y. Lafont and P. Taylor. *Proofs and types.* 1989. Page 55 ff. [Girard et al., 1989]

or
$$\forall x \in |[\mathcal{O}, M, \phi]| \; (\forall y \in a \; \forall P \in Voc(\mathcal{O}) \text{ s.t. } \{x, y\} \subseteq \phi(P)) \; \Rightarrow \; x \in a$$

These two clearly express pretty different, quite opposite!, ideas of maximality. The first one ($\exists P$) considers a loose compatibility and produces a maximal clique in the sense of the largest, most numerous one. The second one ($\forall P$) considers a very strong compatibility, thus producing the clique of maximally compatible elements, which however is likely to be very small, poor of elements, possibly empty. By the way, a concept that gathers all the *qualities* of the other concepts available in a *world* (the one defined by the ontology) could very well be an "impossible" concept, like we said concerning the empty clique. To be exhaustive, we may observe yet a third possibility

$$\forall x \in |[\mathcal{O}, M, \phi]| \; (\forall y \in a \; \exists! P \in Voc(\mathcal{O}) \text{ s.t. } \{x, y\} \subseteq \phi(P)) \; \Rightarrow \; x \in a$$

which obviously produces the maximal clique (in the sense of largest, most numerous) with respect to any particular predicate symbol of $Voc(\mathcal{O})$. However, the most natural among these three is the first one, which is simply existentially quantified, since it preserves the basic idea on which is defined the relation of compatibility.

5.2.1. Interpretation

Coming back to our Ontological Compatibility Spaces we can see a sort of shift concerning what is interpreted in such spaces. Indeed, the aim is to interpret an ontology by an OCS $[\mathcal{O}, M, \phi]$, and a concept (or relation) of this ontology by a clique of $[\mathcal{O}, M, \phi]$. Thanks to such a shift, the basic elements of the web[10] are grouped together by the compatibility relation reproducing, over that piece of "reality" captured in the web, the structure of world specified and logically defined within the ontology $\mathcal{O}$ – thus operating that which one might call the application of an ontology. The groupings by cliques, of the elements on which the ontology is applied, approximate the concepts (and relations) defined in $Voc(\mathcal{O})$. Or, the other way around, concepts are instantiated into sets of objects. Moreover, one must note that such objects cannot be partial. They are the resources in M, they are true (although virtual!), total objects. The fact that we still accept the empty clique $\emptyset$ as a clique is precisely because just such a clique can approximate an impossible concept. Due to so different objectives and aims about what to interpret by means of these spaces (coherent spaces on the one hand and ontological compatibility spaces on the other) one might reasonably ask why do we do such a shift, and why do we still want to recall to coherent spaces while dealing with ontologies, concepts and concrete (virtual) resources? First of all, it is easy to recognize in every concept of an ontology $\mathcal{O}$, and more generally in every predicate symbol of $Voc(\mathcal{O})$, the equivalent of a type in logical sense. They can even be complex types, resulting from the combination of primitive types by means of logical operators – so as to produce complex, defined concepts. Then, one might ask why do we count one space for a whole ontology instead of keeping a coherence space for every concept/-type. A first part of the answer may be that we want to emphasize the local dimension

[10] Here and in next lines web is just the graph resulting once that the compatibility relation has been recorded between points of the support M.

of "applied ontologies" in the World Wide Web. An applied ontology, indeed, is not just an abstract collection of types; it is a specific place in the geography of the Web, a space where resources are given the meaning that makes them identifiable and usable by other applications, services, agents. The collection of objects that provides the support of the OCS gets organized, logically structured by the compatibility relation, which in turn actually produces a logical space with certain characteristics. We have now demonstrated that such characteristics are the same as the characteristics of standard coherent spaces. Therefore we are allowed to conceive and consider directly the "operations" on ontologies (and more generally on information repositories in the Web) as operations on coherence spaces. That is: we gain a calculus to describe and account for information exchange through ontologies.

5.3. Reasonable compatibility

Let's now consider a special class of Ontological Compatibility Spaces: the class of the OCSs such that the theory $\mathcal{T}(\mathcal{O})$ is decidable. For short we could just say that it is the class of the OCSs induced by *decidable knowledge bases.* In the general case, based on our broad conception of KB (as we have introduced it in Definition 4), we cannot guarantee for decidability of the KBs since, for instance, we might be working on an OWL-Full ontology, which is clearly undecidable. As a consequence we restrict the kind of knowledge bases that we will accept in this new special class of OCSs, in such a way that these are only decidable knowledge bases. We may refer then to the branch of Semantic Web that openly relies on the results coming from the area of Knowledge Representation and Knowledge Engineering, and therefore we focus directly on the DL-compliant ontologies, like those written in any of the five dialects of OWL that implement some DL. More precisely, we consider that:

Definition 8 (Decidable Knowledge Base) *A decidable knowledge base is a KB $\langle \mathcal{O}, M, \phi \rangle$ such that the theory $\mathcal{T}(\mathcal{O})$, containing the terminological axioms of the T-box and the assertions of the A-box, is decidable and, in particular, the predicate of derivability from that theory, noted $\vdash_{\mathcal{T}(\mathcal{O})}$, is recursive.*
We may also characterize the knowledge base itself by the recursive derivability predicate by noting the decidable KB as $\langle \mathcal{O}, \vdash \rangle$ [11].

Indeed, we have that ontologies designed based on a Description Logic surely match this condition, since DLs are designed precisely to guarantee decidability of the reasoning on the ontology and the knowledge base that it describes. A DL ontology $\mathcal{O}$ with a non-empty A-box contains already within its signature $S(\mathcal{O}) = \langle Voc(\mathcal{O}), Ind(\mathcal{O}) \rangle$ the set M of resources that the ontology is to describe: it is $M = Ind(\mathcal{O})$ (cf. p. 143). Moreover, it is noteworthy that the recursivity of the derivability predicate is testified by some reasoner that typically is (or at least can be) plugged to any such ontology (cf. p. 144). These are the reasons why we synthetize the notation of decidable knowledge

[11] For sake of simplicity we use just $\vdash$ instead of $\vdash_{\mathcal{T}(\mathcal{O})}$ since the (theory that is the expansion of the) ontology with respect to which the decidability holds is already apparent in the notation $\langle \mathcal{O}, \vdash \rangle$.

bases as $\langle \mathcal{O}, \vdash \rangle$, with the ontology $\mathcal{O}$ providing both T-box and A-box, and $\vdash$ signalling the reasoner capability to decide the knowledge base.

We can now introduce another definition of the compatibilty relation that structures OCSs in case of decidable KBs, one that looks very interesting.

Definition 9 (OCSs – decidable version) *Based on the* decidable *knowledge base* $\langle \mathcal{O}, \vdash \rangle$*, we define the OCS* $[\mathcal{O}, \vdash]$ *induced by the KB as follows:*

- *its support* $|[\mathcal{O}, \vdash]| =$

$$\begin{array}{rl} = & \{x \,|\, x \in Ind(\mathcal{O}) \wedge \exists P \text{ s.t. } \langle \mathcal{O}, \vdash \rangle \vdash P(x)\} \\ \cup & \{\langle x, y\rangle \,|\, \langle x, y\rangle \in Ind(\mathcal{O}) \times Ind(\mathcal{O}) \wedge \exists R \text{ s.t. } \langle \mathcal{O}, \vdash \rangle \vdash R(x, y)\} \end{array}$$

  for P and R respectively unary and binary predicate symbols in $Voc(\mathcal{O})$

- *the compatibility relation between the points of $[\mathcal{O}, \vdash]$, noted $\frown_{[\mathcal{O},\vdash]}$ is assigned according to the following:*
  - *for $x, y \in Ind(\mathcal{O})$*

$$x \frown_{[\mathcal{O},\vdash]} y$$

    if and only if there is a unary predicate symbol P of $Voc(\mathcal{O})$ such that

$$\langle \mathcal{O}, \vdash \rangle \vdash P(x) \text{ and } \langle \mathcal{O}, \vdash \rangle \vdash P(y)$$

  - *for $\langle x, z\rangle, \langle y, w\rangle \in Ind(\mathcal{O}) \times Ind(\mathcal{O})$*

$$\langle x, z\rangle \frown_{[\mathcal{O},\vdash]} \langle y, w\rangle$$

    if and only if there is a binary predicate symbol R of $Voc(\mathcal{O})$ such that

$$\langle \mathcal{O}, \vdash \rangle \vdash R(x, z) \text{ and } \langle \mathcal{O}, \vdash \rangle \vdash R(y, w)$$

  - *and also, for $\langle x, z\rangle \in Ind(\mathcal{O}) \times Ind(\mathcal{O})$*

$$x \frown_{[\mathcal{O},\vdash]} \langle x, z\rangle,\ z \frown_{[\mathcal{O},\vdash]} \langle x, z\rangle \text{ and } \langle x, z\rangle \frown_{[\mathcal{O},\vdash]} x,\ \langle x, z\rangle \frown_{[\mathcal{O},\vdash]} z$$

With an abuse of language, we will refer sometimes to OCSs like this by calling them decidable OCSs*. Moreover, we will call* positive *such OCSs that show the concepts of the corresponding decidable KBs.*

Once redefined the compatibility relation on the basis of the derivability of a statement concerning points of the compatibility space, we can find a meaningful and illuminating realization of that which is normally considered for coherence spaces, but that could sound somewhat puzzling with respect to ontologies and knowledge bases. That is the notion of duality, materialized in the space dual to the one that is considered.

Indeed, given a coherent space X with the coherence relation assigned between the elements of its support, it is immediate to consider the dual space $X^{\perp}$, which has exactly the same support and is structured by the relation of *incoherence*, assigned to every pair of elements of the support that are not coherent in X – except for reflexivity. Now, if we have to consider the same for OCSs we meet some difficulty to conceive a dual ontology (or dual knowledge base) whose representation in form of OCS puts together everything that "normally" is incompatible. Nevertheless, thanks to the second definition of compatibility, we can give some sense to such an anti-ontology (or anti-knowledge base).

Definition 10 (Dual OCS) *The dual of an OCS* $[\mathcal{O},\vdash]$*, that is* $[\mathcal{O},\vdash]^{\perp}$*, is noted* $[\mathcal{O},\nvdash]$ *and defined as follows:*

- *its support is the same as that of* $[\mathcal{O},\vdash]$*, built based on the same set of resources* $Ind(\mathcal{O})$
- *the compatibility relation between the points of* $[\mathcal{O},\nvdash]$*, that is the incompatibility from the viewpoint of* $[\mathcal{O},\vdash]$*, is noted* $\frown_{[\mathcal{O},\nvdash]}$ *and is assigned according to the following:*
  - *for distinct* $x, y \in Ind(\mathcal{O})$
  $$x \frown_{[\mathcal{O},\nvdash]} y$$
  if and only if for every unary predicate symbol P *of* $Voc(\mathcal{O})$
  $$\langle\mathcal{O},\vdash\rangle \nvdash P(x) \text{ or } \langle\mathcal{O},\vdash\rangle \nvdash P(y)$$
  - *for distinct* $\langle x,z\rangle, \langle y,w\rangle \in Ind(\mathcal{O}) \times Ind(\mathcal{O})$
  $$\langle x,z\rangle \frown_{[\mathcal{O},\nvdash]} \langle y,w\rangle$$
  if and only if for every binary predicate symbol R *of* $Voc(\mathcal{O})$
  $$\langle\mathcal{O},\vdash\rangle \nvdash R(x,z) \text{ or } \langle\mathcal{O},\vdash\rangle \nvdash R(y,w)$$
  - *and also, for* $\langle x,z\rangle \in Ind(\mathcal{O}) \times Ind(\mathcal{O})$
  $$y \frown_{[\mathcal{O},\nvdash]} \langle x,z\rangle \text{ and } \langle x,z\rangle \frown_{[\mathcal{O},\nvdash]} y$$
  for every $y \in Ind(\mathcal{O})$ *distinct from* x *and* z*.*

We will call negative *the OCSs that represent non-concepts of decidable KBs.*

One may note in particular that compatibility in $[\mathcal{O},\vdash]$ (i.e. $\asymp_{[\mathcal{O},\vdash]}$) is *large*, whereas its dual, the incompatibility, that is the compatibility in $[\mathcal{O},\nvdash]$ (i.e. $\frown_{[\mathcal{O},\nvdash]}$) is *strict*. In other words, it requires incompatible points to be distinct – which appears quite reasonable, since it would need a "strange animal" to be incompatible with itself . . .

From the dual OCS $[\mathcal{O},\nvdash]$ we can recover the same things that are represented in $[\mathcal{O},\vdash]$ but from the opposite point of view. That is, for any given predicate $P \in Voc(\mathcal{O})$, $[\mathcal{O},\nvdash]$ provides us with the clique of the objects that are not together within the concept denoted by P. Also, by switching polarity we change the engine that provides inference, and then computes (in)compatibilities according to the last definitions of OCSs. That is, from the inference engine ($\vdash$) we pass to the "non-inference" engine ($\nvdash$). It is not the engine that computes anti-theorems, since that which $\nvdash$ finds are not theorems. Nor they are theorems of a theory complementary to that of the original decidable knowledge base $\langle \mathcal{O}, \vdash \rangle$. They are not at all theorems. Rather, $\nvdash$ finds (and shows over the web of its space) that which is not theorem of $\langle \mathcal{O}, \vdash \rangle$.

Remark 3 *We observe that every clique that we can find in the OCS* $[\mathcal{O}, \nvdash]$ *– dual to the OCS* $[\mathcal{O}, \vdash]$*, both representing the decidable knowledge base* $\langle \mathcal{O}, \vdash \rangle$ *– must be:*

- the clique of all the objects in $Ind(\mathcal{O})$ that are mutually incompatible with respect to the concept denoted by some predicate symbol of $Voc(\mathcal{O})$;
- or the clique of all the objects in $Ind(\mathcal{O})$ that are mutually incompatible with respect to some *new* concept not denoted by any predicate symbol of $Voc(\mathcal{O})$;
- the empty clique $\emptyset$, that corresponds to a sort of "inevitable" concept;
- following the inverse direction, for every predicate symbol $P \in Voc(\mathcal{O})$, the OCS $[\mathcal{O}, \nvdash]$ shows the clique of all the objects which are incompatible with respect to P.

Briefly, on the one hand the cliques of a positive OCS (the one that shows compatibility) approximate concepts of the corresponding ontology – that is the part of the formal theory embedded in the knowledge base that is represented by the OCS. On the other hand the cliques of the negative OCS (the one that shows incompatibility) approximate non-concepts, that is collect objects that cannot be put together to instantiate any concept of that ontology. It must be noted however that in both cases compatibility and incompatibility depend on a particular definition of the world, i.e. the ontology. The inverted representation of a knowledge base provided by dual OCSs is mostly appreciable on maximal and empty cliques. In the positive OCS we had room enough to consider three alternative ideas of maximal clique, here we can do the same. Let us see all them together. Having in mind that, for Definition 10, $x \frown_{[\mathcal{O}, \nvdash]} y$ if and only if $\langle \mathcal{O}, \vdash \rangle \nvdash P(x)$ or $\langle \mathcal{O}, \vdash \rangle \nvdash P(y)$, based on the different notions of incompatibility (broader or narrower) and the corresponding different ways of quantifying over P, we can have the following three alternatives. Given a clique a, a is maximal in $[\mathcal{O}, \nvdash]$ iff

$$\forall x \in |[\mathcal{O}, \nvdash]| \; (\forall y \in a \; \forall P((\mathcal{O}, \vdash) \nvdash P(x) \vee (\mathcal{O}, \vdash) \nvdash P(y))) \Rightarrow x \in a$$

or

$$\forall x \in |[\mathcal{O}, \nvdash]| \; (\forall y \in a \; \exists P((\mathcal{O}, \vdash) \nvdash P(x) \vee (\mathcal{O}, \vdash) \nvdash P(y))) \Rightarrow x \in a$$

or even

$$\forall x \in |[\mathcal{O}, \nvdash]| \; (\forall y \in a \; \exists! P((\mathcal{O}, \vdash) \nvdash P(x) \vee (\mathcal{O}, \vdash) \nvdash P(y))) \Rightarrow x \in a$$

of which, the first one ($\forall P$) produces a maximal clique according to the idea of maximally incompatible elements – but anyway it is clearly the largest, most numerous clique in $[\mathcal{O}, \nvdash]$ such that all its objects are mutually incompatible for any considered *quality* – that is *essence* of a concept. The second one ($\exists P$) produces a maximal clique according to the idea of the weakest incompatibility between its elements: it is enough to be incompatible with respect to some *quality*. This reading is likely to produce the largest, most numerous clique, but is based on a notion of incompatibility not so interesting. The third one ($\exists! P$) is quite interesting since it allows to build the largest clique of elements incompatible with respect to one specific *quality*. Moreover by computing all such cliques – the maximal non-concept corresponding to any predicate symbol in $Voc(\mathcal{O})$ – and then operating by progressive intersection between cliques, up to the last remaining clique we get back to the largest possible clique that contains the objects maximally incompatible (the one resulting from the first definition). Nevertheless, as we noted above, the largest possible such clique might even be the empty clique. The non-concept represented indeed has to be the absence of every quality defined in the ontology, so that the best approximation tends to no object. In fact we named it “inevitable” concept while describing the empty clique in negative OCSs. It is inevitable as far as we look at its intentional meaning: every other concept contains it. It is inescapable since, before specifying other qualities, every concept is like that. It is precisely a *bottom* concept. However, the most natural among these three is the first one, which is universally quantified, since it preserves the basic idea on which is defined the relation of incompatibility. Moreover, it is just the dual formulation of the most natural definition of positive maximal clique.

We may keep on in comparing positive and negative notable cliques by remarking that in a positive OCS, maximal clique and empty clique stay basically distinct, since there the maximal clique approximates the largest, most numerous clique – which tendentially could be the whole compatibility space – but according to the weakest idea of compatibility (this is specially true of the idea of maximal clique expressed by $\exists P$). As regards the empty clique in the positive space, it approximates an “impossible” concept that can be instantiated by no objects in the space. Yet, the other idea of maximal positive clique (with $\forall P$) tends to such an empty clique, indicating a possible candidate for the role of “impossible” concept. It indeed asks for a concept that is the concentrate of all the qualities defined in the ontology, so that the number of elements satisfying it tendentially decreases along with the increase of the number of qualities.

Remark 4 *Finally, a decidable knowledge base $\langle \mathcal{O}, \vdash \rangle$ induces two ontological compatibility spaces, one dual to the other, noted $[\mathcal{O}, \vdash]$ and $[\mathcal{O}, \nvdash]$ which share the same support, built based on $Ind(\mathcal{O})$, and theory $\mathcal{T}(\mathcal{O})$ but show, respectively, compatibility and incompatibility between the objects of the support by deriving, respectively, theorems and non-theorems of $\mathcal{T}(\mathcal{O})$ and approximating, respectively, concepts and non-concepts of the ontology $\mathcal{O}$.*

Note also that such non-concepts are not the same as the “uninteresting” concepts that can be found in positive OCSs, i.e. those which show the compatibility between resources. Uninteresting concepts indeed are possible concepts for which is not (yet) known a specific term to call them. Non-concepts, on the contrary, are absolutely not concepts: they collect resources that are incompatible – according to the specific relation of incompatibility that is generated within the particular KB considered.

5.4. Example – OCS

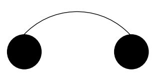
Based on the same ontology of which we have shown some extracts in the first part of this work, we now show what may look like its representation as an Ontological Compatibility Space. We remark that what is going to be shown in an OCS are the actual objects accounted for in the knowledge base, grouped as cliques as long as there is some concept in the ontology of which all the objects in any clique can be proven to be an instance.

We repeat here below the relevant excerpt from the example of part I that we are going to consider:

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/" >
<foaf:Person rdf:ID="me">
    <foaf:name>Marco Romano</foaf:name>
    <foaf:workInfoHomepage rdf:resource="http://logica.uniroma3.it/~romano"/>
    <foaf:mbox rdf:resource="mailto:m.romano@uniroma3.it"/>
    <foaf:knows>
        <foaf:Person>
            <foaf:name>V. Michele Abrusci</foaf:name>
            <foaf:mbox rdf:resource="mailto:abrusci@uniroma3.it"/>
        </foaf:Person>
    </foaf:knows>
</foaf:Person>
</rdf:RDF>
```

This excerpt is an A-box, that contains the following elements of information: two Persons, their Email addresses and the Web-page of one of them, that is five "elementary resources" that populate the base set M of the support of the OCS $|[\mathcal{O}, M, \phi]| \subseteq (M \cup (M \times M))$ as explained before[12]. These elements are introduced with the predicate symbols provided by the FOAF vocabulary (note the prefix before predicate names), so that the terminological part of the knowledge base (the T-box) is recalled *via* the "invocation" of the FOAF namespace. To finish the reduction of this A-box to the most general case of knowledge base $\langle \mathcal{O}, M, \phi \rangle$ one may consider a very simple RDF parser able to read this A-box. Now, the OCS corresponding to such a simple knowledge base shows the following cliques (we illustrate them step by step):

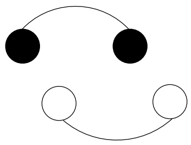
- a clique of Persons



It counts two points because in the few lines of the example A-box we had just two resources bearing the label `foaf:Person`

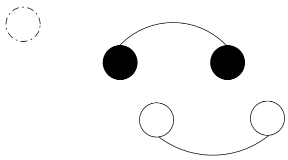
[12] Note also that we present this A-box as part of a generic KB – not a decidable KB – since actually the T-box that is coupled with it does not offer any guarantee about decidability: it was originally an RDF schema, subject later to an attempted conversion in OWL not completely ultimated and, most of all, not bound to DL dialects. Nevertheless, to be honest, most of KBs that one may find in the (Semantic) Web would produce just generic OCSs.

- and a clique of Email accounts



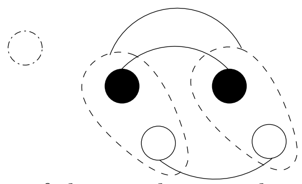
Again two points as in the "code" above we have two strings introduced by the label `foaf:mbox`

- and a clique of Web-pages



It counts just a single point since in the example A-box there is only one string with the label `foaf:workInfoHomepage`

- and also a clique of pairs made of a Person and his Email account



This clique counts two points (dashed circles in the figure) because in the A-box we have two pairs made of a resource labelled `foaf:Person` and a string attributed to it by the relation `foaf:mbox`

In spite of the simplicity and poverty of such example, we additionally remark that concepts in OCSs, as cliques, look just like completely connected subgraphs. Moreover the relations defined in the ontology let one take pairs of points instead of single points as resources.

5.5. Operations on OCSs

The best that the interpretation of ontologies as OCSs may bring to Semantic Web is the consequent interpretation of "operations" on ontologies as operations on such spaces, thus recovering (part of) the "calculus" of coherence spaces for use in Semantic Web. All that may provide, through further development of the basic ideas that we pose in next chapter, a theoretical frame to describe the use of ontologies as sources of information on the part of semantic web-agents.

As we have seen in the previous chapter, usual operations with ontologies range from segmentation (module extraction) and extension to union of ontologies and ontology merging, passing through the family of mapping-based operations. Among these, just ontology union is really an operation, whereas for all the others we have provided a sort of definition based on the specific relation in which the involved ontologies are posed by each of those processes. After all we had noted that all such process can be reduced to just a few basic mechanisms such as inheritance and mapping, which indeed underly the other processes, and, most of all, they can be seen as the two moments of the logical rule of *cut*: the discovery of corresponding points – say two equivalent concepts for instance – on which the cut pivots, and the flowing of information concerning those points from a structure to the other – access to data of that type, or sharing of knowledge about the concepts (their logical definitions). By the way, we just recall that the *cut* is precisely the logical rule for communication between proofs.

On the other hand, coherence spaces interpret all the operations available within the logical system of LL:

- a pair of binary additive operations (*with* & and *plus* $\oplus$), dual to each other
- a pair of binary multiplicative operations (*tensor* $\otimes$ and *par* $\parr$), dual to each other they too
- another binary multiplicative operation: linear implication, noted $\multimap$
- and three unary operations such as *negation* (noted by means of $^{\perp}$), which switches the polarity leading from a space to its dual (positive to negative and negative to positive), and the exponentials (*of course* ! and *why not* ?) which allow respectively to switch from additive (where resources are subject to consumption and therefore finitely disposable within a process that uses them) to multiplicative situations (where resources are reusable as many times as one needs), and *viceversa*.

As a consequence we have some "new" operations to consider with respect to ontologies (via their use on OCSs). It is worth further studying them as to identify their possible role in describing actual interactions (information exchanges) between agents in the Web.

The first operation to be considered is the negation, the switch to the dual coherence space, since we have already seen how to account for duality with ontological compatibility spaces:

$$[\mathcal{O}, \vdash]^{\perp} = [\mathcal{O}, \nvdash]$$

Negation indeed causes the switching between the positive and negative version of an OCS in an involutive manner. With Coherence spaces this implies that cliques from two spaces dual to each other cannot share but singletons. With OCSs we meet this condition and then have an involutive negation provided that one of the three alternative criteria to read maximal cliques in OCSs is chosen and equally used to look for approximants of concepts (and non-concepts) in both spaces (cf. p. 149 and p. 154). Indeed, depending on the quantifier used over the predicates that are to assign compatibility between points of the OCS, one may look at:

- the broadest-but-weakest (in)compatibility (when $\exists$ is used);
- the strictest (in)compatibility (when $\forall$ is used);
- the (in)compatibility specific to any given concept (when $\exists!$ is used).

Moreover we observe that there is no priority between the positive and negative version of an OCS, since both can be directly induced by the decidable knowledge base.

We may note now that this operation does not correspond to any standard operation or process on ontologies. Not even the computation of non-theorems, showing non-concepts, computed by the non-inference engine of a negative OCS (structured by the incompatibility relation marked by $\nvdash$) can be associated to optimization techniques, heuristics possibly adopted to improve algorithms for inferential engines by computing anti-theorems – since the non-inference engine does not derive anti-theorems, that is theorems of a complementary theory. It just computes non-theorems (cf. Definition 10).

5.5.1. Additive operations

The operations performed by means of & and $\oplus$ are called *additive* because they operate on the web that is the sum of those of the spaces involved. Over the resulting web, then, they produce different assignments of compatibility. That is, given two OCSs $[\mathcal{O},\vdash]$ and $[\mathcal{O}',\vdash]$ the web of their & and $\oplus$ will be

$$|[\mathcal{O},\vdash]| + |[\mathcal{O}',\vdash]| = (|[\mathcal{O},\vdash]| \times \{1\}) \cup (|[\mathcal{O}',\vdash]| \times \{2\})$$

In other words, the webs are coloured so as to guarantee that their elements stay distinct and then are stitched side by side. Now, on such a web & and $\oplus$ distribute compatibility in different ways. The following table shows how. In order to improve readability, the compatibility is just noted $\supset$ and the columns say which is the relevant OCS.

$[\mathcal{O},\vdash]$	$[\mathcal{O}',\vdash]$	$[\mathcal{O},\vdash]\ \&\ [\mathcal{O}',\vdash]$	$[\mathcal{O},\vdash] \oplus [\mathcal{O}',\vdash]$
$a \supset b$		$a \supset b$	$a \supset b$
	$c \supset d$	$c \supset d$	$c \supset d$
$a, b \in |[\mathcal{O},\vdash]|$	$c, d \in |[\mathcal{O}',\vdash]|$	$a \supset c, a \supset d,$ $b \supset c, b \supset d$	

Looking at the table the discourse is quite simple: in the resulting web, both the operations & and $\oplus$ preserve compatibility of the points that are compatible in one of the original space. Moreover, $\oplus$ adds no additional compatibilities between points of $|[\mathcal{O},\vdash]| + |[\mathcal{O}',\vdash]|$, whereas & assigns compatibility also to every pair made of one point from $[\mathcal{O},\vdash]$ and the other from $[\mathcal{O}',\vdash]$. To put it in a nutshell, $[\mathcal{O},\vdash]$ and $[\mathcal{O}',\vdash]$ are compatible according to the & and incompatible according to the $\oplus$.

With respect to the OCSs, it is easy to see that $[\mathcal{O},\vdash] \oplus [\mathcal{O}',\vdash]$ provides exactly the same results that one would expect from the only real operation on ontologies, the union of ontologies. It indeed merely takes all the cliques of $[\mathcal{O},\vdash]$ and all the cliques of $[\mathcal{O}',\vdash]$ and just puts them into the new web $|[\mathcal{O},\vdash]| + |[\mathcal{O}',\vdash]|$. But on the contrary we have absolutely no correspondent on the part of ontologies for the dual operation $[\mathcal{O},\vdash]\ \&\ [\mathcal{O}',\vdash]$. This last indeed, besides the same cliques as the $\oplus$, takes also for granted – “irresponsibly” – the compatibility between every clique of $[\mathcal{O}, \vdash]$ with any clique of $[\mathcal{O}', \vdash]$. Such a mixing up not only has no meaning with respect to ontologies, but if we try to follow that which it causes for the OCSs, we note that it just weakens the original ontologies by creating new cliques that approximates concepts not defined nor identified by any predicate symbol neither in $Voc(\mathcal{O})$ nor in $Voc(\mathcal{O}')$.

Nevertheless, the & becomes really interesting if we use it – which is the dual of $\oplus$ – on the OCSs dual to $[\mathcal{O}, \vdash]$ and $[\mathcal{O}', \vdash]$, that is if we consider

$$[\mathcal{O}, \nvdash] \,\&\, [\mathcal{O}', \nvdash]$$

Such a switch of polarity allows us to keep on considering, along with the positive version of the union of ontology, also its negative version, that is the point of view on the non-theorems and non-concepts of the ontology $\mathcal{O}$ together with those of $\mathcal{O}'$ – exactly in the same way as we do with a single ontology that can be observed from the dual points of view of the positive and negative OCSs that it induces. In particular we observe that the assignments of incompatibility seem to be more reasonable now: the cliques of $[\mathcal{O}, \nvdash] \,\&\, [\mathcal{O}', \nvdash]$ are all the non-concepts approximated in $[\mathcal{O}, \nvdash]$, in $[\mathcal{O}', \nvdash]$ and finally also every non-concept approximated by cliques spreading over these two OCSs, basically assuming that they are incompatible.

Remark 5 *We conclude then that the additive operations $\oplus$ and & on OCSs correspond to the operation of union of ontologies, respectively in their positive and negative representation as OCSs:*

$$\mathcal{O} \uplus \mathcal{O}' \text{ induces } [\mathcal{O}, \vdash] \oplus [\mathcal{O}', \vdash]$$

and

$$\mathcal{O} \uplus \mathcal{O}' \text{ induces } [\mathcal{O}, \nvdash] \,\&\, [\mathcal{O}', \nvdash]$$

5.5.2. Multiplicative operations

The operations performed by means of $\otimes$ and $\invamp$ are called *multiplicative* because they operate on the web that is the product of those of the spaces involved:

$$|[\mathcal{O}, \vdash] \otimes [\mathcal{O}', \vdash]| = |[\mathcal{O}, \vdash] \invamp [\mathcal{O}', \vdash]| = |[\mathcal{O}, \vdash] \multimap [\mathcal{O}', \vdash]| = |[\mathcal{O}, \vdash]| \times |[\mathcal{O}', \vdash]|$$

Over the resulting web, then, they produce different assignments of compatibility. In other words, the webs are joined and compatibility is assigned according to the following:

$$\begin{array}{rcl}
(a, b) \frown_{[\mathcal{O}, \vdash] \otimes [\mathcal{O}', \vdash]} (a', b') & \Leftrightarrow & a \frown_{[\mathcal{O}, \vdash]} a' \;\wedge\; b \frown_{[\mathcal{O}', \vdash]} b' \\
(a, b) \smallfrown_{[\mathcal{O}, \vdash] \invamp [\mathcal{O}', \vdash]} (a', b') & \Leftrightarrow & a \smallfrown_{[\mathcal{O}, \vdash]} a' \;\vee\; b \smallfrown_{[\mathcal{O}', \vdash]} b' \\
(a, b) \frown_{[\mathcal{O}, \vdash] \multimap [\mathcal{O}', \vdash]} (a', b') & \Leftrightarrow & a \frown_{[\mathcal{O}, \vdash]} a' \;\Rightarrow\; b \frown_{[\mathcal{O}', \vdash]} b' \\
 & \wedge & a \smallfrown_{[\mathcal{O}, \vdash]} a' \;\Rightarrow\; b \smallfrown_{[\mathcal{O}', \vdash]} b'
\end{array}$$

Let us try to give directly an interpretation of all this with respect to operations on ontologies:

- the $\otimes$ provides us with the coupling of every approximant of a concept (a clique) of $[\mathcal{O}, \vdash]$ with every approximant of a concept of $[\mathcal{O}', \vdash]$. Clearly there is no process that uses ontologies to produce anything like that. Nevertheless, one might define the vocabulary $Voc(\mathcal{O} \otimes \mathcal{O}')$ such that within the OCS $[\mathcal{O}, \vdash] \otimes [\mathcal{O}', \vdash]$ there are all the approximants of the concepts defined in it – or alternatively one may think of that as the extraction of a sub-space of $[\mathcal{O}, \vdash] \otimes [\mathcal{O}', \vdash]$ which contains just those meaningful and interesting (approximants of) concepts;

- the $\multimap$ provides us with a joined space whose cliques couples together an approximant of a concept of $[\mathcal{O}, \vdash]$ with an approximant of a concept of $[\mathcal{O}', \vdash]$ whenever the first (approximated) concept implies the latter. With a little flexibility we can see here precisely the mechanism of ontology mapping, although here it looks like "overabundant" since there is nothing that specifies when such an implication should occur, so that every concept of $[\mathcal{O}, \vdash]$ implies any concept of $[\mathcal{O}', \vdash]$. However, as for the previous case, we may consider the theory $\mathcal{T}(\mathcal{O} \multimap \mathcal{O}')$ such that within the OCS $[\mathcal{O}, \vdash] \multimap [\mathcal{O}', \vdash]$ there are all the meaningful and interesting implications between approximants of the concepts defined in $\mathcal{O}$ and $\mathcal{O}'$. Finally, it must be noted the second part of the definition of the $\multimap$ operation, which poses a noteworthy constraint: distinct objects (approximants) must be mapped into distinct objects (approximants), due to linearity (and stability on approximants) of the $\multimap$. By the way, this last condition looks like easily fulfilled by knowledge bases, since any form of data mapping requires that every single value (object, resource, individual) stay distinct, though altogether collected under a common type-name (the concept name for instance);

- as regards the $\parr$, there is no much ground to attempt a comparison with operations on ontologies. Nevertheless, $\parr$ is the dual operation to $\otimes$, so that we will consider it as the alternative point of view on it:

$$([\mathcal{O}, \vdash] \otimes [\mathcal{O}', \vdash])^{\perp} = [\mathcal{O}, \nvdash] \parr [\mathcal{O}', \nvdash]$$

  where one can see incompatibility ($\frown_{[\mathcal{O}, \nvdash] \parr [\mathcal{O}', \nvdash]}$) instead of compatibility.

Remark 6 *We may say that the multiplicative operations ($\otimes$ and $\multimap$) somehow represent on OCSs the mapping-related operations on ontologies. In particular:*

- *an OCS resulting from $[\mathcal{O}, \vdash] \multimap [\mathcal{O}', \vdash]$ contains* all possible mappings *between two ontologies – that is inclusions of the (approximants of) concepts of $\mathcal{O}$ in (approximants of) concepts of $\mathcal{O}'$;*

- *an OCS resulting from $[\mathcal{O}, \vdash] \otimes [\mathcal{O}', \vdash]$ contains all possible couplings of the (approximants of) concepts of two ontologies $\mathcal{O}$ and $\mathcal{O}'$. In other words, it contains the* results *of every possible mapping between those ontologies.*

The $\multimap$ seems to deal with the instructions of mapping. Just to give the idea, this operation on OCSs may be considered as providing an interpretation of the *query* that operates the mapping, thus something closer to the *map* function of the previous chapter, that operates at the intensional level of ontologies. Yet, this operation provides all possible such queries, and therefore "hides" among many useless queries the few queries that would do the job for a meaningful mapping operation between ontologies. The $\otimes$ on the contrary produces the actual join of data that results from a mapping, something closer to the *rea* process of the previous chapter, that operates at the extensional level of ontologies. It approximates the new concepts that appear after a mapping specification is given, though, again, it "hides" the interesting concepts among too many uninteresting concepts.

5.5.3. Limits of these correspondences

We do not even try to introduce exponential operations on OCSs. For the time being we are not able to conceive any interesting reading of such operations as operation on ontologies. Nonetheless, we imagine that they would deserve attention later on, possibly after that some theory based on Ludics has provided a model for interaction of web agents so that exponentials could play a role in controlling the use of resources by agents – we will attempt in next chapter to (just) open this way.

All the considerations on correspondences between operations on OCSs and "operations" on ontologies are just an attempt to shed some light on the possible advantages of using the theory of Coherence spaces to describe that which happens while operating on ontologies. As the ontology union is the only real operation on ontologies that we have found in the previous chapter, so only the union ($\oplus$) of OCSs has a clear well defined meaning with respect to ontologies or, more generally, knowledge bases. However, in the case of an OCSs like $[\mathcal{O}, \vdash]$, also the dual of the union (i.e. &) becomes as meaningful as the union, since they actually do the same operation, just from opposite points of view. Indeed, in the decidable case, one can switch between positive and negative interpretation of an ontology and also of its concepts, thanks to decidability of the $\vdash$ and the involutivity of negation. Once chosen one of the three possible readings of *maximal clique* that we have presented above (cf. p. 154 for the case of OCSs representing decidable KBs) concepts, and non-concepts, can be found in OCSs as such maximal cliques. The same then holds for the operation of union, with the possibility to shift between its positive ($\oplus$) and negative (&) account.

On the contrary, this is not possible in the general case of OCS, like $[\mathcal{O}, M, \phi]$, for in the absence of a reasoner that finds concepts and non-concepts of the ontology (more generally theorems and non-theorems of the corresponding theory) it would be somewhat innatural to consider its dual OCS.

The multiplicative operations, on the other hand, do not exactly correspond to any operation or process on ontologies. They would need additional conditions and restrictions to properly model real processes of data mapping or ontology merging, precisely like ontology mapping (and all mapping-related processes) and merging need special techniques and *ad hoc* algorithms – together with knowledge about the world external to the (formal) ontology – to produce meaningful results. Nevertheless, through the observation of multiplicative operations on OCSs, we may gain interesting insights on how to logically model the exchange of information over the (World Wide) Web. Most of all, in the dynamic of mixing cliques together from two different spaces (as with $\otimes$)

we can see in action that which we called the “second moment” of the fundamental logical rule of *cut*, that is the flow of information between logical structures. It is, in the practice of the Web, the moment when communication is enabled and data from different data sources get “mixed” in a new common environment – though clearly data may be still stored in distinct physical places since the mixing is produced at the level of logical representation.

From an ontological perspective however, operations like these are not innocent, for they cause some transformation of the original concepts of each ontology involved. Once a concept C from ontology $\mathcal{O}$ is mixed with a concept C' of ontology $\mathcal{O}'$, none of both is the same concept as before. Typically, the idea lying behind the concepts is weakened, allowing for a larger number of objects (points of the OCS, resources) to be counted as compatible. And also the notion of compatibility itself, though represented by the usual mathematical relation, is a new one, since actually the exact notion of compatibility that is embedded in any ontology is also specific to any given OCS. We mean here, for “notion of compatibility”, the overall idea according to which knowledge engineers design an ontology; in other words, the intensional knowledge of the domain that they try to record in an ontology. In a nutshell: the ideas lying behind the conceptualization formalized in the ontology. We may note, then, that additive operations, on the contrary, preserve the original notions of compatibility – and, in the case of the &, also add a new one that concerns both ontologies as wholes and make them mutually, globally, compatible.

Anyway, we still have nothing to say concerning the “first moment” of the *cut*, that is when it is discovered, when two objects or concepts are recognized as being equivalent[13] and, because of their equivalence, susceptible to activate communication between the ontologies that they come from, or by which they are described. We take into consideration this issue, among other things, in the next chapter, where Ludics will be introduced as an instrument for agents to “logically talk” about what they already know (something like loaded ontologies) and possibly discover new knowledge.

For the time being we conclude our presentation of OCSs with respect to ontologies by defining their relationship with the operations that we have just examined.

Theorem 1 *The class of* positive *Ontological Compatibility Spaces* $[\mathcal{O}, \vdash]$*, induced by decidable knowledge bases, is closed under the operation of union (*$\oplus$*) on Coherence spaces. The class of* negative *Ontological Compatibility Spaces* $[\mathcal{O}, \nvdash]$*, induced by decidable knowledge bases, is closed under the operation dual to the union (*&*) on Coherence spaces. The negation (* $^{\perp}$*) relates these two classes allowing to switch from one to the other, and viceversa, from every OCS to its dual, and viceversa.*

Proof. We may divide the proof in three steps, one for each operation involved:

[13] In typical logical settings, in order to establish a *cut* one would rather look for dual objects. However here it is just a matter of properly defining the context: imagine of a web agent looking for some information from a website. It will fulfill its mission when, say, some website will have given it the information that the agent was seeking, not the dual of that information. It is the different attitude in searching vs providing, asking vs answering that which introduces “polarities” in the picture.

union $\oplus$: given two decidable OCSs $[\mathcal{O},\vdash]$ and $[\mathcal{O}',\vdash]$, their union $[\mathcal{O},\vdash] \oplus [\mathcal{O}',\vdash]$ is yet an OCS that represents some Knowledge Base, namely, the KB $\langle \mathcal{O} \uplus \mathcal{O}', \vdash\rangle$ resulting from the disjoint union (as defined in the previous chapter, p. 126) of two ontologies. In particular, this KB is the union of the theories originally presented in each ontology separately, and it is still a consistent theory for the Craig Interpolation theorem. Every interrogation on the resulting OCS $[\mathcal{O},\vdash]$ $\oplus$ $[\mathcal{O}',\vdash]$ will receive an answer from either the reasoner plugged to $[\mathcal{O},\vdash]$ or from the reasoner plugged to $[\mathcal{O}',\vdash]$, with no risk for "clashing" answers;

dual of the union $\&$: given two OCSs $[\mathcal{O},\nvdash]$ and $[\mathcal{O}',\nvdash]$, induced by decidable KBs $\langle\mathcal{O},\vdash\rangle$ and $\langle\mathcal{O}',\vdash\rangle$ looking at the incompatibility between resources, the dual of their union, that is $[\mathcal{O},\nvdash]$ $\&$ $[\mathcal{O}',\nvdash]$ is yet an OCS that represents the same Knowledge Base as the previous case, i.e. $\langle \mathcal{O} \uplus \mathcal{O}', \vdash\rangle$, but from the point of view of incompatibility (cf. Definition 10). In particular, the structuring relation (incompatibility) will be, correctly, extended to any pair of points a, b such that $a \in |[\mathcal{O},\nvdash]|$ and $b \in |[\mathcal{O}',\nvdash]|$ (cf. p. 159);

negation : from the definition of negation (cf. p. 158) and of dual OCS (Definition 10) it is apparent that the negation of an OCS like $[\mathcal{O},\vdash]$, i.e. $[\mathcal{O},\nvdash]$, still represents the same decidable knowledge base, just from the opposite point of view, where compatibility has been replaced with incompatibility.

$\square$

As regards general OCSs, the ones induced by Knowledge Bases that do not offer special warranty about decidability of the reasoning, we may consider a milder form of closure:

Proposition 2 *The class of Ontological Compatibility Spaces like $[\mathcal{O}, M, \phi]$ induced by (generic) knowledge bases is closed under the operation of union ($\oplus$) as defined for Coherence spaces, which corresponds to the operation of union of ontologies.*

Proof. Let $[\mathcal{O}, M, \phi]$ and $[\mathcal{O}', M', \psi]$ be two generic OCSs and $[\mathcal{O}, M, \phi] \oplus [\mathcal{O}', M', \psi]$ their union. Based on the definition of the union of OCSs (cf. p. 159) and of the process of union of ontologies (cf. p. 126) we can see that

$$[\mathcal{O}, M, \phi] \oplus [\mathcal{O}', M', \psi] = [\mathcal{O} \uplus \mathcal{O}', M \oplus M', \phi \oplus \psi]$$

where:

$\mathcal{O} \uplus \mathcal{O}'$ has been already defined (cf. p. 126)

$M \oplus M'$ is the union of the collections of resources M and M'

$\phi \oplus \psi$ is the function that maps predicate symbols from the language of $\mathcal{O} \uplus \mathcal{O}'$ into sets of resources in $M \oplus M'$, more precisely from $Voc(\mathcal{O}) \oplus Voc(\mathcal{O}')$ to $(\mathfrak{P}(M) \cup (\mathfrak{P}(M) \times \mathfrak{P}(M))) \oplus (\mathfrak{P}(M') \cup (\mathfrak{P}(M') \times \mathfrak{P}(M')))$, defined thus:

$$\begin{aligned} \phi \oplus \psi(P) &= \phi(P) \text{ if } P \in Voc(\mathcal{O}) \\ &= \psi(P) \text{ if } P \in Voc(\mathcal{O}') \end{aligned}$$

And it corresponds with no doubt to the Knowledge Base produced by the process of union (as described at p. 126) of the KBs $\langle\mathcal{O}, M, \phi\rangle$ and $\langle\mathcal{O}', M', \psi\rangle$. $\square$

Obviously also the other operations, the multiplicative $\otimes$, $\parr$ and $\multimap$, can be performed on OCSs, but we have no warranty concerning the correspondence that they may have with any possible Knowledge Base, thus weakening the idea itself of OCS. In fact, concerning the $\otimes$ for instance, in the product of two OCSs we can recover the arrangement of cliques corresponding to the results of a mapping, but together with too many other mixed cliques that do not correspond to any meaningful mapping of concepts. Similarly, $\parr$ will have no significance either. Even less in the absence of a reasoner – that is in case of an OCS like $[\mathcal{O}, M, \phi]$ – that makes interesting the notion of incompatibility of dual (negative) OCSs. And it is yet a little harder to conceive $[\mathcal{O}, \vdash] \multimap [\mathcal{O}', \vdash]$ as a genuine OCS, since it rather provides something like approximants of mapping instructions – whereas cliques in $\oplus$ and $\otimes$ directly approximate concepts (though too many concepts).

5.6. Example – operations on OCSs

Let's now have a quick look to what is in practice the representation of the operations on ontologies as operations on OCSs. We tackle here just the two cases of $\oplus$ and $\otimes$, that correspond respectively to ontology union, as the sum of concepts, and (roughly) to ontology mapping, as the coupling of the concepts in an ontology $\mathcal{O}$ with any concept of another ontology $\mathcal{O}'$, although a useful and meaningful mapping on ontologies would typically associate a concept in $\mathcal{O}$ with just one concept in $\mathcal{O}'$.

In order to show that which happens when operating on OCSs we use again the A-box of section 2.4 and the new one here below, which is based on another ontology, similar to FOAF, for personal information management (PIM) [Berners-Lee et al., 2000], developed by a small group of researchers involved in the W3C's Semantic Web Initiative[14]. It is absolutely minimial, but it is enough to give the idea of how it all might work.

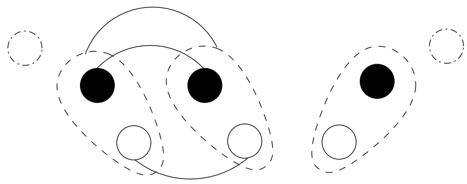
```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:pim="http://www.w3.org/2000/10/swap/pim/contact#">
   <pim:Person rdf:ID="cf">
      <pim:name>Christophe Fouquere</pim:name>
      <pim:mailbox rdf:resource="mailto:cf@lipn.univ-paris13.fr"/>
      <pim:homePageAddress rdf:resource="http://www-lipn.univ-paris13.fr/~fouquere/"/>
   </pim:Person>
</rdf:RDF>
```

[14]The ontology is available on-line at http://www.w3.org/2000/10/swap/pim/contact. Among the people who have been working on it since early 2000s let us mention Tim Berners-Lee, co-inventor of the WorldWideWeb.

It introduces three elementary resources (a person, his mailbox address and his webpage) that we easily recognize as perfectly correspondent with the types of data contained in the other knowledge base, but there is no evidence for a computer to argue like that. Let us call $[\textsc{pim}, M', \phi']$ the OCS induced by this KB, and $[\textsc{foaf}, M, \phi]$ the other one. First of all we consider the simplest (trivial) way to put together the information within two different knowledge bases, i.e. what we have called their sum ($\oplus$):

$$[\textsc{foaf}, M, \phi] \oplus [\textsc{pim}, M', \phi']$$

As the figure shows, facts asserted in the two ontologies are just put side by side, no mixing of information can take place with such operation. The result is simply to have all the cliques from the two OCSs collected in a new single OCS.

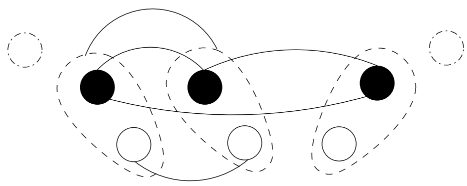


Then we consider the more interesting operation of mapping. We recall that this operation aims to establish equivalence between concept names (more generally speaking: predicate symbols) specified in different ontologies whenever they are intended to describe the same concept. How to discover when actually two symbols, two concept names may be used for the same concept is the crucial point, but for the present example we will rely on the most general case of

$$[\textsc{foaf}, M, \phi] \otimes [\textsc{pim}, M', \phi']$$

We do not show a picture to illustrate the distribution of compatibilities since it would be hardly readable. It is easier to understand by words for this time: in the $\otimes$ space all the cliques of $[\textsc{foaf}, M, \phi]$ are compatible with any clique of $[\textsc{pim}, M', \phi']$. That is a situation that surely contains also the particular case for which we give the following picture.

Here the mixing of cliques affects only the cliques corresponding to the labels `foaf:Person` and `pim:Person` even though, reasonably, if one considers the actual meaning of the other facts described in the two A-boxes, also other cliques could (meaningfully) mix (i.e. those corresponding to `foaf:mbox` and `pim:mailbox`, and the same for `foaf:workInfoHomepage` and `pim:homePageAddress`).



Such a distribution of compatibility is quite meaningful and it depends only on a specific axiom that we had in the FOAF specification reported in the first part of this work, and that we now repeat just for the relevant excerpt:

```
<rdfs:Class rdf:about="http://xmlns.com/foaf/0.1/Person"
```

```
      rdfs:label="Person" rdfs:comment="A person.">
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
   <rdfs:subClassOf>
   <owl:Class rdf:about="http://www.w3.org/2000/10/swap/pim/contact#Person"
      rdfs:label="Person"/>
   </rdfs:subClassOf>
</rdfs:Class>
```

To put it in a nutshell, this is a mapping specification for the predicate `foaf:Person` with respect to the Pim ontology, on the specific predicate `pim:Person`. Such a mapping instruction is embedded in the specification of the FOAF ontology, so that one may also consider this as a plain case of inheritance. Nevertheless, also the mapping works in the same way: once provided the mapping specification we just have to see inheritance come into play (which manifests the "second moment" of the *cut*). What stays mysterious in our account of operations on ontologies as operations on OCSs – we already noted it – is the first moment of the *cut*. More precisely: there is no general logico-mathematical means to account for the production of the special vocabulary of the "multiplied" OCS ($[\textsc{foaf}, M, \phi] \otimes [\textsc{pim}, M', \phi']$) such that it restricts the set of cliques to be observed over the web of this OCS; such that it provides the list of interesting and meaningful concepts to be observed in that OCS.

5.7. OCSs and Social Web

We are going now to illustrate that which we consider as the real value added of our proposal for representing knowledge and information in the World Wide Web. Indeed, *per se* the OCSs may be just an original (maybe singular) technical exercise from which one, in the best case, can gain some interesting insights concerning ontologies, knowledge bases and the like. On the contrary, we are pretty convinced that OCSs can be also really useful to study and model interaction in the "other half" of Semantic Web, that is Social Web. Yet before facing issues like the interaction between web agents, we think that the huge amount of data that are produced, recorded and used in many Web2.0 services could largely benefit from a theoretical account that does not "pervert" their dynamic nature by forcing the reduction to a fixed, statically taxonomical yet basically flat hierarchy, like a poor relative of an ontology. We think that precisely the OCSs have something to offer also, and maybe most of all, to such kinds of data sources like, for instance, folksonomies, since they provide a very general and flexible theoretical framework, yet endowed with sufficient logical rigour.

Nowadays, indeed, when looking at the set of tag-terms adopted within a community, it is expected to reconstruct a formal ontology out of that, establishing a neat and formal hierarchy among concepts, useful for resource retrieval according to the traditional top-down approach of progressive specification. The major side effect of such a reduction is the loss of the dynamic aspect of Web2.0.

We may observe that building the Semantic Web by means of ontologies requires predefined sets of metadata (the schemes[15]) to be adopted and respectfully obeyed. Their

[15]Official documents concerning Web standard languages at W3C use both the plural forms schemes

usefulness – and the wealth of Semantic Web itself as the workplace of autonomous agents – will depend in fact on the number of resources whose set of metadata matches one or more of the predefined schemes so that programs specially written according to the same scheme(s) will be able to use those resources. So the Web of data[16] that W3C indicates turns out to be something like a giant database where a neat definition of the logical scheme (even composed of many different ontologies) can be achieved only thanks to the standardization of the metadata tags to be used to describe resources.

In the opposite direction goes the practice of free tagging, so that folksonomies emerge as everlasting works in progress where concept-terms institution and resource description and classification always happen in the same time, with no hope for standardization. In fact, when people tag they freely choose and establish their own categories in an unending process of ontology elaboration. Moreover, while using their very personal categories people also express their own "world's understanding" so that tagging spaces are not only useful for classification, but also convenient for collective intelligence to share knowledge.

We remark that tagging spaces publish enormous amounts of resources with some kind of classification while providing a cognitive framework (a structure in which to collect, compose, record and share information) that has not the claims of ontology but is powerful enough to let one recognize and find classes of resources that are compatible, i.e. similar, better: equivalent to some extent, under some respect.

Perhaps such a cognitive framework is a lower quality contribution, with respect to formal DL ontologies, to have the content of the Web surely recognizable, but it seems to follow a more feasible way to achieve that. In fact, since it does not depend on the technology of given-in-advance semantics – no standard set of tags nor consensus of domain experts is needed in order to tag resources – it preserves the dynamical behaviour of Web2.0 and, precisely for that, it could render the Semantic Web a "common people affair". This participation by common people, which has powered Web2.0, has also been the reason of the enrichment of the Web not only as a place for business but also for knowledge (in a very broad acception) exchange during last years. And at the same time, is getting louder the voice that asks for collaboration between communities of Semantic Web and Social Web in order to reach for that which somebody already (improperly) calls Web3.0. Clearly it is not a matter of versioning, but the point is to really and effectively join together the efforts conducted by researchers on the technical and theoretical part (Semantic Web) with the natural evolution of the Web and the social activity of human users that spontaneously partecipates in building the Web (Social Web).

It is in this perspective that we think that OCSs can be somewhat useful. Instead of the usual techniques for tags clustering and concept extraction, like FCA, finalized to recover some (poor) ontology, we may rather exploit OCSs in order to recover from tagging spaces a description of the resources contained in a datasource that is formal enough to be useful for data exchange but that does not need *ad hoc* specification of a conceptual hierarchy. We look only for compatibility between resources observing the

and schemas, whereas the most correct, or at least formal schemata occurs almost never.

[16] As it is written as the first definition of Semantic Web at "its home page" – http://www.w3.org/2001/sw/.

connections given within an OCS. Briefly, our proposal is to give more logical dignity to flat tagging spaces without rising too high in formal complexity so as to prevent large contribution from common web-users. We simply aim to describe flat spaces in such a way that it makes sense to talk about operations between them.

In defining the OCSs we relied on a very general notion of knowledge base, which indeed we have defined by means of the triple $\langle \mathcal{O}, M, \phi \rangle$, where $\mathcal{O}$ is the linguistic part, used to read concepts and draw cliques, M is the "concrete" part and ϕ is the connection between them.

Now, in order to have an OCS out of a tagging space, that is the basement of a folksonomy, we need nothing more than what has been stated for ontologies. Going into OCSs we could even consider folksonomies as very simple ontologies, if folksonomies would not had one dimension more than ontologies, that is users. Indeed, in the previous chapter we have defined a folksonomy as the union ($\bigcup$) of the personomies of which it is made of. Let us start then with personomies. A personomy $\mathcal{P}_{u \in \mathsf{U}} = \langle \mathsf{T}_u, \mathsf{R}_u, \mathsf{A}_u \rangle$ with $u \in \mathsf{U}$ to recall the user dimension, provides us with an extremely simple knowledge base, whose linguistic part is just the set of tags T_u used by user u, whose concrete part is the set of resources R_u tagged by u and whose evaluation ϕ might be just a parser that reads the personomy and records the assignments in an extremely simple theory. The support for the OCS induced by the personomy, based on the definition of OCS, should be some set, that for the time being we note simply X, such that $X \subseteq (\mathsf{R}_u \cup (\mathsf{R}_u \times \mathsf{R}_u))$ and more precisely

$$X = \{x \mid x \in \mathsf{R}_u \wedge \exists P \text{ s.t. } x \in \phi(P)\} \cup \{\langle x, y \rangle \mid \langle x, y \rangle \in \mathsf{R}_u \times \mathsf{R}_u \wedge \exists R \text{ s.t. } \langle x, y \rangle \in \phi(R)\}$$

with P and R predicate symbols (actually tags) in T_u. But since in T_u – like in the whole T by the way – there is no binary tag[17] we take directly R_u as the support. From this point on the OCS $[\mathcal{P}_{u \in \mathsf{U}}, \mathsf{R}_u, \phi]$ induced by the personomy works exactly as any other general OCS. In particular, the compatibility relation is assigned between the points of the support whenever two objects appear together within the same concept, that is in terms of folksonomies, whenever two resources bear the same tag.

Finally, a folksonomy is the sum of all the personomies available in some Web2.0 service, that we consider indexed by the set U of users. Therefore, the OCS corresponding to the whole folksonomy is

$$[\mathcal{F}_{\mathsf{U}}, \mathsf{R}, \phi] = \bigoplus_{\mathsf{U}} [\mathcal{P}_{u \in \mathsf{U}}, \mathsf{R}_u, \phi]$$

with $\mathsf{R} = \bigcup_{\mathsf{U}} \mathsf{R}_{u \in \mathsf{U}}$ and, we just recall, $\mathsf{R}_{u \in \mathsf{U}} = \{r \mid \exists t \text{ s.t. } \langle r, t \rangle \in A_{u \in \mathsf{U}}\}$ for some user u and tag $t \in T_{u \in \mathsf{U}}$. Or to say it in a simpler way, the OCS representing the folksonomy is the logical sum (the operation $\oplus$ on OCSs) of all the personomies generated by the users of the service.

[17] It may be argued that binary tags are not science fiction and sooner or later we could find them largely used in Web2.0 applications. Nevertheless it is noteworthy here just the simplest case (only unary tags), since binary tags would take us back exactly to the same situation already described for general OCSs.

Just one remark concerning the interpretation of folksonomies. Cliques in the OCSs approximate concepts defined in the theory which is the "linguistic" part of the knowledge base represented in the OCS itself. Now, within a personomy the approximation of concepts by cliques is not only an arrangement in order to account for the particular set of data collected in the knowledge base. Rather, it is surprisingly similar to what happens when a concept (the idea behind the tag-term used to classify resources) is continuously altered, usually enlarged in order to accommodate some new resources – or possibly tightened. Every addition of resources within a concept causes some weakening of the idea of compatibility that is shared by all the resources that bear the tag before the new resources get it.

If we extend the discourse to the whole folksonomy, we must note that its representation as the "big" sum of personomies produces an OCS that is suitable to undergo further operations, as well as any single personomy that composes the folksonomy can participate in other operations, so that it could be very interesting to consider for instance the mapping between personomies, or the merging of full folksonomies.

6. Logic, Web and Interaction

In the previous chapter we have introduced a theoretical model for the representation of ontologies and folksonomies[1] – that of Ontological Compatibility Spaces (OCSs) – that provides a common logical interpretation of data sources designed according to any of the two alternative lines of activity in Semantic Web: i) the line that follows from XML and RDF and considers just how to richly represent data structures, and ii) the line that bridges these efforts with the experience, and logico-technical results, of Knowledge Engineering (e.g. those which design and use DL-compliant OWL ontologies).

In the present chapter, then, we will introduce our *first attempt* to set up a theoretical model for the representation of the interaction of the web agents that should share and exchange between them the information and knowledge contained in all those kinds of information repositories. After some clarification of the forms and ways of interaction that we can at present envisage within the scenario of Semantic Web, we will often speak in the sequel of *web agents' interaction*, or even *communication* or simply *interaction*, while always meaning this same issue. That is, we will be always concerned with the use, on the part of (special) web agents, of information and knowledge made available and accessible in the Web. Preliminary step will be to clarify which kind of web agents we are considering.

Moreover, as OCSs are based on objects borrowed from Linear Logic, the Coherence Spaces, which are at the same time richer structures than sets (of Set theory) but also more flexibile – that which makes them good candidates for providing the logical representation of a variety of forms of knowledge (and information) representations –, in a similar way we base our proposal for modelling web agents' interaction on other objects (called *designs*) borrowed from another theory (Ludics) that comes from Linear Logic.

Whereas the advantages of using OCSs to represent knowledge are, mostly, in the possibility to deal with ontologies and folksonomies (and the like) in the Web with the same theoretical means and on a common ground for clean logical structuring of the information and knowledge contained therein, the advantages of adopting a Ludics-style representation of interaction between web agents will be again in the generality of the approach, but also in some additional benefits that Ludics may bring for a deeper understanding of the notions of compatibility and of concept in OCSs, together with original insights to consider ontology itself from the point of view of interaction – something like an interactive way of defining concepts. In order to show all this we will first present Ludics and its main achievements through a quick introduction to the subject that does not aim to be exhaustive but sufficiently clear and informative

[1] And, in principle, of any data source in the Web designed according to the fundamental recommendations of the WorldWideWeb Consortium to promote interoperability of systems and share of data

about all the aspects that we will exploit in proposing our model of (Semantic) Web interaction – as well as informative about the other aspects that, though, stay critical with respect to this proposal.

The reliance on Ludics, indeed, is not just a technical choice determined by the previous choice to adopt (and adapt) Coherence Spaces to provide the logical representation of information in the Web. Actually, to be honest, at the present stage of our research there is yet something to be fixed in order to recover, within the special context of the Web and for the use and the purpose that we aim to, the precise relationship that originally holds between Coherence spaces and Ludics *designs*. Nevertheless, as OCSs appear useful to tackle some difficulties with the logical interpretation of knowledge and information representations available in the Web, so Ludics seems able to tackle other issues equally important concerning the way to use that information.

Therefore, our choice to adopt (and adapt) Ludics to describe the communication between web agents depends mostly on other reasons, which are shared, for the most, with other two interesting, recent, attempts to adopt Ludics and take advantage of it for applications external to proof theory. The first one is the use of Ludics to interpret, describe and model human communication in natural language, as it is proposed in [Lecomte and Quatrini, 2009]. But there is also another promising attempt that is concerned directly with the Web. It deals with Web in general, as opposed to Semantic Web specific issues that are at the core of the present work, and focuses on inter-operations between web-servers and web-clients, that is the basis of the World Wide Web functioning (cf. [Fouqueré, 2011]). Fouqueré introduces a functional programming language (FICX – *Functional Interactive and Compositional XML*) capable to model the client-server communication as a dialogue, by adopting the logical framework provided by Ludics. Such a language is intended to cope in a reliable and logically grounded way with a number of issues and problems that mark the common experience of Web navigation, even on the part of basic users.

The fundamental assumption with that language is about the opportunity to represent and model Web operations as dialogues between servers and clients. After all, the main shortcoming of the usual web languages is in the fact that they are programming languages conceived according to the "stereotype" of computer science, that is as software running on a (single) machine, for which therefore communication is just the exchange of (final) output between servers and clients, with no attention paid to the peculiarities of the Web environment. Yet, the Web is an open environment, where the interaction between the actors involved in each communicative exchange should be more deeply considered, at least for it poses a range of "new" issues like the ones that FICX tackles.

Our attempt, therefore, will follow a similar line, focusing on the peculiarities of the interaction between web agents that share knowledge by means of ontologies and recover (provide) information from (to) a variety of data sources. Their interaction over the Web, indeed, is a matter of communication. A communication that, by the way, is conveyed in a "hybrid" language. Indeed, it is totally artificial, its syntax is perfectly formalized and rigorous, as it should be for any language to be used by machines. But its lexicon is largely unpredictable, since it depends on the many different *vocabularies* (as we have introduced this term in part I of this work) that are provided by different subjects (organizations for standardization, as well as research institutes or business companies) – and possibly used by yet other different subjects – so that its semantics stays largely under-determined, even though logical constraints are given to interpretation by means of ontologies (cf. Guarino's definition of ontology, 54). Basically, this is the problem with Semantic Web. And we believe that basically this problem corresponds to the fundamental problem of communication between any two speakers of a given language, which actually cannot really speak the same language – their underlying ontologies too will always have some, though minimal, difference. With an extreme simplification: as people have their own "private" (implicit) ontology, so web systems and agents have their own special (formalized) ontologies, and communication may succeed inasmuch as there are parts of such ontologies which are shared, and the larger are the shared parts, the higher is the probability for a successful communication.

In this chapter, therefore, we will show why and how (Semantic) Web could benefit from Ludics. Firstly we will consider the forms that interaction between web agents may actually take in Semantic Web, involving ontologies, folksonomies and whatever we may represent as an OCS. Secondly we will briefly introduce Ludics (the ideas and fundamental achievements) and will illustrate to which extent it may be helpful to represent and model web agents' communication. Then, we will present in practice the interpretation of this communication based on a "calculus" that is just a revision of Ludics. We will also show how and to which extent this may corroborate the idea of representing ontologies and, more generally, Knowledge Bases (as defined in the previous chapter, p. 144) as OCSs. Final remarks will signal the open issues that we could not solve within the research period of this PhD.

6.1. Semantic Web interactive processes

Whereas in the previous parts of this work we have outlined the architecture of the Semantic Web, and how it is expected to be achieved based on the original project of the Semantic Web Initiative, here we are going to discuss that part of Semantic Web that has been payed much less attention – not only in the present work (until now) but generally on the part of people working for Semantic Web. That is the part concerned with the actual use of the knowledge and information made available in the frame of Semantic Web. Semantic Web indeed should enable machines to find and autonomously process the data made "understandable" to them (i.e. not simply machine-readable) by means of the network of ontologies (and of other similar artefacts like RDF schemas) wherein are defined the terms used to describe the meaning of those data, their value as information. Such a network is the *web of ontologies* (about which we have told at p. 29) that should become the ground for Semantic Web agents to act on.

Our aim is therefore to sketch here what the operations and processes specific to the Semantic Web should look like, under which conditions they should take place, how they should go over, when they may be considered successfully completed, between which actors and relying on which assumptions they should happen . . . Note however that we must say "should" since in fact all these aspects stay somewhat uncertain in the roadmap of World Wide Web Consortium toward a working Semantic Web. On the one hand the idea itself of Semantic Web agents is pretty woolly and tends nowadays to be confused with that of Web Services, so that it needs to be somehow "restored" in order to properly consider what the agents are to be. On the other hand, it seems to be there so few literature that tackles the specific issue of Semantic Web agents and their specific activity, albeit on the contrary there is a vast literature on intelligent, autonomous software agents. In fact, as we will show in the following, most of the ideas lying behind ontologies and Semantic Web languages have been almost entirely recovered from the results of 1990s researches at the boundary between Artificial Intelligence and software engineering – before the shift in interest toward the World Wide Web – which itself envisaged the development of *software agents* to support some forms of collaborative software engineering by dividing large and complex tasks into smaller, simpler tasks to be carried out by such intelligent agents. Indeed, as [Genesereth and Ketchpel, 1994] testify, the idea itself of software agents was born as an answer to the problems with interoperatio between systems, the major of which is clearly heterogeneity – of programs, of programming languages and of policies with each single software (e.g. as regards the consequences of changes in subsequent versions of a program). They explicitly state that

> *Agent-based software engineering* was invented to facilitate the creation of software able to interoperate in such settings. In this approach to software development, application programs are written as *software agents*, i.e. software "components" that communicate with their peers by exchanging messages in an expressive *agent communication language*. [Genesereth and Ketchpel, 1994]

Based on this observation, for which we will provide further clues, we may plainly accept that exactly that same idea of intelligent agent is expected to come into play in Semantic Web in the vision of W3C. A conviction that, by the way, explains the scarcity of works devoted to define the special dynamics of Semantic Web agents' interaction: it shall be the same dynamics envisaged for the intelligent agents as they have been conceived in 1990s.

As a consequence, in order to recover, for our discourse, the necessary elements that must enter the picture – so that we may properly attempt to model Semantic Web agents' interaction from a logical, theoretical, not algorithmic perspective – we rely on a few crucial texts, a little aged (they date 1990s) but that come precisely from that research movement on intelligent agents. In these texts however we will also find the clues to track the development of techniques for intelligent agents' cooperation (basically Knowledge Representation and Engineering matter) from the 1990s setting (where the aim was to make communicate single specialized systems designed to operate alone and share some output) up to the original setting of the World Wide Web, where the final, overall aim is to make understandable to a new generation of agents the information and knowledge spread in it. In tracking such an evolution of the (notion of) intelligent agents we will also attempt to signal a radical shift of the expectations about what is to be interaction among (and "around") informative systems. Indeed, we believe that there is also another shift that goes along with it, one that calls for a much deeper involvement of philosophy – and philosophers too – to work side by side with systems engineering – and engineers.

For the time being let us start with the definition of Semantic Web agents. Subsequently we will sketch the basic tasks that such agents will have to perform. In doing this, we have to identify as many as possible elements and aspects that may characterize agents' communication in order to have our theoretical interpretation as rich as possible; but at the same time, we have to pose minimal constraints so that our model of interaction can be meaningful with respect to any actual implementation of a protocol for Semantic Web agents communication.

6.1.1. Which agents?

We have introduced initially, in the first part of this work, the agents that should operate in (and exploit) the Semantic Web as *intelligent* and *autonomous web agents*. Subsequently we have called them *web agents tout court*. It is time to distinguish more accurately. First of all we give that our discourse is about software agents, thus excluding other forms of artificial agents (robot, mobot and whatelse), but insisting on the artificial nature of our agents, that "live" (run) in some computerized informative system.

In order to do a clean work, first of all we will reject the adjective *intelligent* which is extremely "slippery", especially after its usage in Artificial Intelligence. We may just signal that intelligent, in the expression *intelligent agent*, suggests that the agent should be able to perform some task (hard to some extent) in such a way that its behaviour, as seen from the viewpoint of the results that it produces, would cause the man of the street to say "That's clever". After all, it seems to be the same for AI in general.

Let's tackle then the other adjective: autonomous. An article by Franklin and Graesse ([Franklin and Graesser, 1997]) in 1996 made autonomy the real characteristic of software agents. After a classification of the software agents developed until then, they ended up with a definition that they deemed formal enough:

> An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future. [Franklin and Graesser, 1997]

The formality of such a definition is arguable, nonetheless it has the merit of signalling autonomy by highlighting the *situation* of the agent and its ability to modify the environment wherein it acts, beyond the simplistic acceptation of autonomy as the ability to (start to) do something for "its own sake" – which in fact, this last, is a characteristic that would not be so significant for Semantic Web agents, and also likely to stay unmatched since those agents are supposed to act on behalf of their masters (the human users) so that their activity should always be, at some time, initiated by a user's request. Hence, we may consider the other aspects of agents' autonomy (their being situated within an environment and the ability to modify that environment as a result of their actions) as relevant also to the case of Semantic Web agents. In particular, in a first approximation, the environment can be identified with the above recalled *web of ontologies*, and the situation of each agent may be defined as the portion of such a web with which the agent is acquainted; the ability to modify it, then, can be identified with the ability to discover new knowledge, that is to discover new portions of the web, to learn new knowledge. These characteristics however still do not say the most interesting part in our opinion. The title of the article from which the quotation above is taken is somewhat more interesting to us: "Is it an agent or just a program?". It matches our repeated interest in the possibility to use information over the Web without *ad hoc* solutions, that is purposely written programs that perform their tasks by unfailingly following some rigid communication protocols that have been provided by some programmers. But what happens when attempting to make communicate systems for which no mapping specification has yet been defined? No agent among the ones proposed till now has any means to cope with such a situation – therefore, no communication at all. The point of [Franklin and Graesser, 1997] seems to be close to ours: to speak of autonomous agents should call for something more than a traditional program fully pre-determined as regards in its possible actions. This is just an executor. But after all, most, if not all, of the agents conceived or abstractly considered at the time of the research on agents for software engineering were indeed just executors, parts of larger systems specifically designed to deal with communication issues with other systems, and consequently – and coherently with the engineering approach – designed looking for working solutions, practical and effective, but strictly focused on making interact two (or a few more) particular systems. Typically the agents were – and nowadays too they are – designed to work in a well defined frame, between well specified systems. In other words they are common programs that make other programs (and systems) communicate. But, in their design, the Web is not such an important dimension of their acting – it is not considered as the *environment* of autonomous (inter)action – rather it is considered just as the place were the agents execute their tasks. Again, we may note incidentally that this is not an error, or a "fault". Rather, it is a pretty natural result, as it derives from systems engineering, developed according to an engineering perspective that aims to make things to work.

Nevertheless, the advent of Semantic Web provides us with a slightly, but remarkably different setting. We are going to appreciate the main differences by rehearing what was said in 1990s concerning the obstacles and difficulties that the World Wide Web posed (then) to the use of agents in it.

Our next reference is then Petrie ([Petrie, 1996]). After tidying up the discourse on software agents (basically by deeming useless both the adjectives intelligent and autonomous referred to such agents), Petrie conducts – already in 1996 – a clean distinction between the agents developed as research projects by AI research groups and the Web-based agents, that were then developed almost exclusively by commercial companies. He found these last, apart from being actually somewhat less "intelligent" than the former, to be not really agents, but something else. Of course, he admitted that in the absence of a generally accepted definition of software agent everything could be named like that. Nevertheless, he notes, based on the actual tasks executed by those Web-based agents, they would have been better called *servers* – thus being also perfectly compliant with the well established jargon and style of the World Wide

Web – whereas only marketing reasons would have justified the abuse to call them agents. Indeed, those 1990s Web-based agents accomplished – maybe very well and by means of some tricky interconnection of different databases, to compose a distributed database – just the services of a common Web server that "answer carefully formatted questions" ([Petrie, 1996] p. 26). Again, this is the kind of activity that we may find still today happening in the Web and that we do not want to count among the activities of Semantic Web agents – since they pertain to the traditional approach to make systems exchange data. As Franklin and Graesse would say, that is just another program, not an agent.

On the contrary, we may recover from [Petrie, 1996] a couple of aspects that characterize "true" agents. These are

- the ability to co-operate in order to accomplish their tasks,
- and the ability to "think back" on earlier queries.

Whereas the first one is said like that by Petrie – who indeed insist on the possibility that agents exchange messages not only while firing queries and receiving the answers, but also, for instance, when looking for who (which other agents) may help it to accomplish its task[2] – the second one is our more general reading of Petrie's observation about the possibility that a (true) agent may firstly give an answer to some query and later on come back on the same subject by signalling that something has changed concerning the matter that the earlier query was about. Generalizing a little, this signals as a possible requirement for authentic web agents the ability to "keep in mind" previous user's requests and serviceably notify whenever some newly discovered piece of information allows to give a different answer to an already answered request. Generalizing yet a little more this suggests the possibility that agents build up a store of knowledge by collecting pieces of information – clearly this one is the direction towards which we are interested to lead the discourse.

However, Petrie found also two order of hindrances to the achievement of a scenario wherein agents co-operate and accumulate knowledge (according to our reading) on the Web:

- the inadequacy of the HTTP protocol to support such a kind of activities,
- and the inadequcy of the (form of) information in the Web.

Concerning the first problem, Petrie found that the HTTP protocol (HyperText Transfer Protocol), that is the basis of World Wide Web functioning, being a client-server protocol does not support, by its nature, none of the characteristics pointed out here above (agents' cooperation and "rethinking") and, as a consequence, he asks for some kind of peer-to-peer (P2P) protocol to properly support them (think for instance of the co-operation of web agents and in particular of the search for "collaborators", e.g. other agents from which to receive new informations). We agree on this observation,

[2] Petrie says explicitly: "That candidate agents should exchange shared protocol messages to collectively perform a task differentiates agents from simple expert systems or other knowledge-based systems", [Petrie, 1996].

and are also willing to develop our theoretical model of Semantic Web agents' interaction by taking into account the peer-to-peer dynamics. But, on the one hand, since it is mostly a technical matter, the "choice" of the protocol to adopt for dealing with the exchanges of information between agents stays on a level lower than the one where we aim to take the discourse, that is that of the logical comparison, and possibly "stitching", of different knowledge representations (in particular, different knowledge bases). Our theoretical interpretation of Semantic Web agents, therefore, will be somewhat transparent to the protocol issue, though it will be apparent that it could find application on the Web only if supported by a peer-to-peer protocol or, maybe, by the original programming language for Web applications proposed by [Fouqueré, 2011], which is purposely designed to properly deal with interactive processes.

Concerning the second point, on the contrary, we have that things have now changed a lot with respect to the situation of 1990s. Indeed, Petrie lamented that information in the Web was produced and formatted only as a matter to be displayed on computer screens, so that actually web agents had nothing to do with that –

Such a meaningless formatting confirms the idea that Web-based agents were a "bluff", inasmuch as their working was just a matter of clients querying servers; and, on the other hand, the use of the Web just as the place where the "services" take place diminishes the idea itself of web agents. Without an adequate formatting of that information, indeed, the real benefit of working in the World Wide Web – that is most of all the possibility to attain to an incomparable repository of information – is not effectively achievable, at least not by having machines accessing information on our behalf, as web agents should do. But it is precisely the objective of the Semantic Web initiative to give that information a meaning which is understandable by machines, that is by web agents. Let us conclude then this regression in the past of the Web and of software engineering so as to finally show the deep connection between these two worlds, that we may consider to have culminated in the Semantic Web Initiative. If we step back to see what was there at the origin of OWL we do not find only RDF. In fact, besides the work of W3C to promote standards for the Semantic Web, there was yet a previous effort on both the banks of the Atlantic Ocean concerned with ontologies and standard protocol to support interoperation of systems by means of those ontologies. In particular, there were the DAML and OIL projects, from which shortly after has been derived OWL, and especially that part of OWL that implements Description Logics. OIL (standing for both Ontology Inference Layer and Ontology Interchange Language) was the project on the European side. It was mainly concerned with bringing to the Web the results of Knowledge Representation with Description Logics. On the American side, the DAML (DARPA Agent Markup Language) was a project concerned with bringing the ideas about agent-based software engineering to the Web. It came after other experiences of that department of the USA national (military) research that, among other things, had mostly contributed the development of Internet (i.e. the DARPA – Defense Advanced Research Projects Agency). The researches that we allude to refer to an overall long-term project that can be fully compared with the Semantic Web, except for some substantial differences that, we think, depend on the different stage of development of the Web – originally just a place *in* which agents had to execute tasks, and nowadays the environment *with* which agents should interact in order to accomplish their tasks.

Curiously we may find a forceful synthesis of this older DARPA project in the above mentioned paper by Genesereth and Ketchpel ([Genesereth and Ketchpel, 1994]). It illustrates, with great optimism and largely simplifying many issues, how the interoperation among different software systems should be achieved by means of software agents. Without listing here all the elements, we focus on that which is the fundamental component of that picture, that is the *agent communication language*. We have already met it in the quotation from [Genesereth and Ketchpel, 1994] at page 174. Here we recover some other words about that language:

> ACL (Agent Communication Language) can best be thought of as consisting of three parts – its vocabulary, an "inner language" called KIF (short for Knowledge Interchange Format), and an "outer" language called KQML (short for Knowledge Query and Manipulation Language). [Genesereth and Ketchpel, 1994]

It is surprising to consider that in these lines dated 1994 there is, compressed, most of the Semantic Web (two or three of its layers, think for instance of RDF, OWL, and SPARQL) and even more: there is in particular also a quite well defined idea (as it is exposed in the rest of the paper by Genesereth and Ketchpel) about a protocol for agents to exchange messages – that which is still missing in the picture of Semantic Web. It is yet more exciting the position expressed at the end of that article about the scope of such a project on interoperation of systems: "Our long-range vision is one in which any system (software or hardware) can interoperate with any other system, without the intervention of human users or their programmers" ([Genesereth and Ketchpel, 1994]).

But, beyond being suggestive and evocative, this picture appears soon to be too simplistic. It is worth to report original words so as to unveil shortcomings:

> The vocabulary of ACL is listed in a large and open-ended dictionary of words appropriate to common application areas. Each word in the dictionary has an English description for use by humans in understanding the meaning of the word; and each word has formal annotations (written in KIF) for use by programs. The dictionary is open-ended to allow for the addition of new words within existing areas and in new application areas.
>
> Note that the existence of such a dictionary does not imply that there is only one way of describing an application area. Indeed, the dictionary can contain multiple ontologies for any given area. For example, it contains vocabulary for describing three dimensional geometry in terms of polar coordinates, rectangular coordinates, cylindrical coordinates, etc. A program can use whichever ontology is most convenient. The formal definitions of the words associated with any one of these ontologies can then be used by system programs in translating messages using one ontology into messages using other ontologies.

Today, after the experience of Semantic Web, and the quick tour we have had on it till now, we can see that it is not all so nice and easy, and Semantic Web is far from achieved precisely due to the communication issues related to the presence of different ontologies. In spite of the possible accuracy of KIF or whichever alternative knowledge representation formalism (like OWL), to translate from an ontology to another one is all but an easy task. Even less now, since ontologies are no longer to describe a software system produced by professionals from the viewpoint of software engineering; but are to represent a domain of knowledge produced by, and for the use of, a community of quite common people. The main change and fundamental shift with Semantic Web is from sharing information about systems, to sharing systems of information and knowledge.

Paradoxically, one could say that heterogeneity has "infected its cure". That is, the idea of a common language by means of which to make communicate different systems, since it allows for different vocabularies to be "loaded" in the language, has only displaced the problem with heterogeneity at an upper level, with an evident gain as regards the ease to deal with such an heterogeneity – since it is well formalized in ontologies. But after all it is yet there, waiting for *ad hoc* solutions like mappings between any two systems that have to share data and exchange information.

Now, the tight intertwining between Semantic Web and these previous efforts in software agents and Knowledge Representation research means that, if on the one hand Semantic Web is nothing new, on the other hand we may take for granted and consider as still valid much of the work on agents' interactive dynamics from that which has been produced then, in 1990s. The relative scarcity of interest in Semantic Web agents as a specific area of research, after all, seems to confirm the impression that this could be the actual assumption also on the part of the community working for the Semantic Web. Semantic services indeed (or Semantic Web services) appear to be precisely the readjustment of older Web-agents in such a way that they may directly exploit Semantic Web effort in enriching information with metadata. But, according to the series of characteristics pointed out till now, it is quite easy to recognize that, as even their name suggest, these are yet another form of client-server interaction, which relies on servers rather than on real agents.

In our opinion, on the contrary, it is useful to recover older positions about web agents, but it is also important to consider them in the light of the evolution of the Web, and in particular after the phenomena of Web2.0 and the Social Web. So, in next section we will signal the most interesting ideas that could be yet recovered from ACL, besides that which have been already put into Semantic Web; but we will also signal the few aspects that are illustrated – although vaguely – in [Genesereth and Ketchpel, 1994] and that still miss in Semantic Web. At the same time we will also try to propose for these last a reading from our point of view that could explain our reliance on Ludics as the theoretical framework to interpret Semantic Web agents' interaction.

Finally, to conclude with the definition of the kind of agents that are needed for Semantic Web, besides the aspects of autonomy, co-operation and rethinking individuated above, let us consider the definition of agent proposed in [Genesereth and Ketchpel, 1994]:

> The criterion for agenthood is a behavioral one. An entity is a software agent if and only if it communicates correctly in an agent communication language like ACL. This means that the entity must be able to read and write ACL messages, and it means that the entity must abide by the behavioral constraints implicit in

the meanings of those messages.

Excluding the direct reference to the specific language ACL, we want to save the notion of *behavioural criterion*: we will present it in an pretty new understanding after having introduced basic elements of Ludics.

6.1.2. Which processes?

We will focus here in particular on the description of the activity of agents as regards their exploration of the *web of ontologies*, and their learning too. By learning we mean the increase of knowledge after discovering new portions of that web, which should cause some logically significant composition of languages and theories. This is actually a very high level perspective, far from any particular (and perhaps actually useful) implememtation. Nevertheless we think that it could be helpful to design implementations that take into a greater consideration the real nature of Semantic Web interactive processes. Serious implementations of Semantic Web agents, in our opinion, should be concerned with philosophical issues about ontology and communication along with the usual engineering problems, and should benefit from the insights that come from both fields of investigation. Concerning our idea of "learning agents" then, we do not aim, at least within this work, to compare it with any particular machine learning approach among the ones developed in AI research. Indeed, by making no assumption about the algorithmic implementation that actual Semantic Web agents could be given, we prefer no to mention machine learning, as it would call almost automatically for a number of ideas, notions and intuitions that we are not interested in so far. For us, the learning of Semantic Web agents is just the extension of their knowledge by the expansion of their known vocabulary, and as a consequence also of the corresponding logical theories wherein the terms of the vocabularies are defined. It is a learning by expansion of the knowledge base that every agent somehow brings along.

While presenting our reading of web agents interaction we will aim to give the proper consideration to the remarks of Petrie ([Petrie, 1996]) about the peer-to-peer nature of that activity, but we will not insist too long on this, since our primary interest is about the composition of languages, vocabularies and theories – coherently with the work done till now on "operations" between ontologies and Ontological Compatibility Spaces.

Let us now recover, as promised short above, the most interesting aspects of the Agent Communication Language presented by Genesereth and Ketchpel (p. 179). We will just take our inspiration from that, and will provide an original reading that we deem adequate to the current situation of the World Wide Web. Basically, ACL is made of three parts: ACL vocabulary, the logical definition of it (produced with KIF), and the KQML that is the part that actually enable communication: it expresses requests from agents to agents, and the corresponding answers, and also manages the "overhead" – we mean in particular the activity, typical of peer-to-peer protocols, of searching for other agents (the peers) and also the collateral information that wraps the query-answer messages (data about the requestor, timestamps, maybe digital signatures, and the like) that allows to track and make robust the communication.

We are not interested here, within this thesis, in this last dimension of communication – the one concerned with the "overhead". As regards the part of ACL expressed by means of KIF, we may note that within Semantic Web this role is played by OWL, especially in its DL-compliant sublanguages . It is not too clear on the contrary, what should be the ACL vocabulary. If it has to be a sort of top level ontology (think of foundationl ontologies of part I) that every agent should know and every system should be compliant with – as the words "The vocabulary of ACL is listed in a large and open-ended dictionary of words appropriate to common application areas" after all suggest – we must say that this has not (yet?) been achieved in Semantic Web – and we are pretty sure that it has not been achieved in agent-based software engineering either. But we, instead, may consider that it has to be the overall metalanguage that allows to define all the other ontologies (vocabularies) proper to each single, particular system, then we have that, again, this is already there in Semantic Web: it is RDF, and also OWL in general. Finally, what about KQML, apart from overhead management? This is the crucial point to our discourse, and we think that it is also the crucial point in Semantic Web. By the way, Semantic Web misses an adequate query language, and we deem that exactly this one is the reason why Semantic Web is not yet (fully) in service. It misses the key element to enable fruitful interaction between agents. Within ACL, KQML provides the tools to pose queries between agents and get answers back. Actually, we have a query language also for Semantic Web, but we would say that SPARQL (our Semantic Web query language) is just RDF/OWL with question marks. Even though SPARQL is technically much more than that – it closely resembles a database query language like SQL, nevertheless it is stuck to the idea of Web services, like 1990s Web-based agents that really were just client-server services happening over the Web but that had little (or nothing) to do with agents. SPARQL, Semantic Services and the current line of exploitation of Semantic Web provide just a new generation of client-server applications, maybe much easier to compose, but nothing substantially new. It needs always some engineers to "sit at a table" and agree on the languages and how to format information in such a way that it can be successfully shared between systems. Sometimes it happens that some particular language (ontology or vocabulary) gets much popular and gets adopted by many people without any negotiation of languages (like for instance with any of the alternative standards of RSS) simply because it has been accepted as a standard (official or *de facto*), thus enabling a wide range of special implementation of the same service. But it is just a client-server relationship (e.g. with RSS for contents delivery over the Web).

We are not to say that KQML is anything much different from that[3]. Reasonably, it should be the same, since Semantic Web, as we have seen, comes almost directly from the same roots. But it is time now to think something radically new. That is something that makes room for real agents, with the characteristics of autonomy expressed in the previous section, and that allows to exploit the treasures of the Web: the information and knowledge spread in it.

That which we have in mind is a lightweight protocol for agent communication that

[3]We must admit that we are not acquainted with KQML, which however is not so a vital branch of research after the shift to Semantic Web.

provides some standard types of messages, in particular a small set of type of queries that an agent may pose to another agent. The information that should be exchanged between the agents, as the object of their communication by means of such a protocol, is not only the base data that are described by the vocabulary upon which there already exists an agreement. It is also the specification of the vocabulary itself, and the set of standard types of queries should give just the minimal language to ask, for instance, the definition of a term in such a way that the single agent may (autonomously, with no intervention of human experts) match it against the terms that it already knows. Basically, the core idea is this: any two agents may communicate by means of such a lightweight query protocol and tell each other what they know; whenever both they know a given term (which is uniquely identified thanks to its namespace), this will provide the handhold for a *cut*, that is a logical connection between the theories that define the vocabularies of each agent. Learning then, is precisely this mechanism of expansion of known vocabularies (and relevant theories) by the addition of new terms as soon as their definition relies on some already known term. Once an agent learns a new piece of information (a portion of an ontology different from the ones it already knew) the agent extends its awareness of the web of ontologies and is able to acquire and use new data, described by that piece of ontology.

We are convinced that such a mechanism recalls much of the dynamics of communication in general, also in natural language. This is a resaon that made us to look for a comparison with the attempts of [Lecomte and Quatrini, 2009] to understand the semantics of linguistic interaction by studying the dynamics of the dialogue. This is also the idea that brings us to consider Ludics as a powerful instrument to deal with communication as a form of interaction. And this, finally, is also the idea that, in our opinion, may lead to real interaction between Semantic Web agents and to a fruitful exploitation of the knowledge spread in the World Wide Web.

As a consequence, we should consider Semantic Web agents as quite simple programs endowed with some Knowledge Base, made of an initial T-box (corresponding to the portion of the web of ontologies that they are acquainted with "out of the box") and possibly of an A-box too (that is punctual information that is coherent with that T-box). Indeed, according to the model of interaction that we are going to describe, Semantic Web agents are really autonomous agents whose programming code does not tell explicitly to which server to ask for some special information, nor which is the tag (or concept-name) to be used in order to ask for that information in the language (i.e. according to the ontology) of the server-system to which the query should be submitted. This is because by doing like this would mean once again to operate on the basis of a rigid protocol wherein all that needs has already been defined by system administrators and programmers, and the agents actually have nothing to do in autonomy. By the way, such a kind of agents is also, automatically, a general purpose agent, not committed to execute just one or a few well defined, special tasks.

On the contrary, we think that in principle a Semantic Web agent should be able to discover all that it needs on its own, or at least be able to attempt it. Thus, the agents, in our setting, only carry their knowledge (that is their original KB) plus a minimal set of instruction about how to conduct query-answer sessions with other agents (that is the lightweight query protocol mentioned above). Most of all, the agents can extend their knowledge (expand their KBs) by adding pieces of knowledge and punctual information (T-box and A-box formulas respectively to their KBs) after every query session with other agents.

To make the discourse a little more concrete, we attempt to sketch what the lightweight query protocol should provide Semantic Web agents with. In particular, we may signal which are the types of queries that we would like to find in the lightweight query protocol. But we do not intend to list all the possible queries; rather we propose here two basic distinct types of queries, sufficient to classify all them into two major groups:

- queries for data retrieval (from A-boxes)
- and queries for T-box exploration.

Queries for data retrieval are quite similar to what already exists: they allow to retrieve punctual information about individuals mentioned in some A-box, so that by means of this type of queries an agent may directly ask another agent about information described with a vocabulary that is common to both. That is, queries for data retrieval can take place only if the agents have verified that they "speak the same language", at least as regards the special portion of their knowledge that is interrogated by the queries. Actually, such queries could even be produced by using SPARQL: there is no reason to reject what is already there, and it is quite powerful for this kind of interrogations. Its only limitation depends on the fact that it may only ask for data that an agent must already know how to talk about – hence it suits perfectly to our first type of queries. As a consequence, to have an idea of the queries that can be counted in this group, one may just think of the same queries that can be written with SPARQL.

But before data retrieval queries may take place, some other exchanges should intercur between the interacting agents. These may be counted in the number of the queries for T-box exploration. Indeed, the idea of exploration can be illuminating. In order to check whether two agents speaks (at least partially) a common language, they might tell each other which vocabularies they know – e.g. by listing the namespaces that they are acquainted with. This could be seen as the "shakehand" phase of the interactive process. Then, if the agents speak a common language, the first type of queries (for data retrieval) may take place, over the data that are described by that common language. If, on the contrary, they do not speak any common language – beyond the query language itself and the standard (meta)languages of Semantic Web (i.e. RDF, OWL and SPARQL) which could be considered as the basic (and possibly only) instructions originally given to every Semantic Web agent – then it needs that the agents explore each other's T-box, searching for terms whose definition calls for other, common terms. Based on the newly found terms, new light on the meaning of the other terms appearing in the vocabulary of each other agent might be shed. Obviously, exploration queries may be posed even in case there is already a common language between the agents and one agent is interested in expanding its knowledge by attempting to acquire new terms from the rest of the other agent's knowledge.

A few examples of this second group of queries will be useful, since they are the original part in the panorama of Semantic Web, though they are not at all novelties in the field of Knowledge Representation. The point would be, actually, to include such queries as standard constructs in the query language, available for agents to interrogate other agent's knowledge on a logically meaningful basis. Indeed, although the use of reasoners (inferential engines) is deemed not a key element for Semantic Web achievement[4], we think that pursuing *cuts* between ontologies could offer important benefits at the cost of a little effort to bring some form of logical reasoning in the open sea of the Web. This does not mean to put interaction between agents after the bottleneck of reasoning on KB, which is actually problematic and quite bad performing – by the way, many important ontologies are not even formalized in DL-compliant variants of OWL. Rather, we could take advantage of the simple, pre-linguistic, geometric approach to logical representation of information and knowledge that we have firstly proposed with Ontological Compatibility Spaces and that would suit well the interactive processes between agents that we are going to describe by means of Ludics. In this way we would put aside for a while the linguistic meaning of the (label stuck on) tags, we would not consider their intended meaning for a human as it can be "calculated" by means of traditional semantic technologies (which after all deal with Natural Language Processing), and just consider the logical value of groups of compatible objects that we can recognize in an OCS, and that we can handle by means of tags as just handles to pick up and look at resources. Additionally, whenever ontologies with a greater logical value should be available, one could also save their support for more complex forms of reasoning, provided that the reasoning is performed in background, not at runtime for each interactive querying session between agents – like after all it actually works in client-server systems that adopt inferential engines. The results of such reasoning could be stored in the memory of an agent as an extension of its initial KB, and thus be available along with all the rest of its knowledge for any querying session starting after the last update of the reasoner's inference drawing.

Now, typical logical questions that may be counted among T-box exploration queries offered by the query protocol could look very similar to special constructs used to interrogate DL Knowledge Bases. Having in mind that a term C in the vocabulary of an agent is a concept-name in the corresponding ontology, in order to understand C an agent might ask for instance:

- the superconcepts (or subconcepts) of C;
- the least subsumer of a concept C, i.e. the strictest superconcept;
- disjoint concepts with respect to C, i.e. classes of resources that cannot be C;
- the logical definition of C (if any), i.e. the formula that expresses the "admission test" (necessary and sufficient conditions) for a resource to be counted in the class of all the C;
- the list of all the axioms where C appears;
- and many other like these, possibly also involving roles of the ontology.

[4] In fact they are usually considered as useful add-ons within closed environment, but not to be used in the open context of the World Wide Web.

Any of these queries will provide the asking agent with a wider insight of the answering agent's KB, and at each step the same (type of) query can be posed again on another concept (or a different query on the same concept) up to exhaustion of possible questions with respect to the answering agent's knowledge, or up to the discovery of some term known by both agents, based on which other terms can be understood and learnt on the part of the asking agent. For any of these queries the dialogue may either go on at the intensional level, looking at concept names – though only as handles to pick up (possible) resources – or step down at the extensional level asking for the set of resources that bear the tag (concept name) on which the dialogue is pending.

Anyway, the reading that we will give of Semantic Web agents' interaction by means of Ludics allows to deal with logic interpretation with no need to refer to model theory and the like. Precisely by virtue of Ludics – which produces a sort of shortcut in the traditional distinction between syntax and semantics – we (and the agents actually) have just to manipulate the special objects of the Ludics calculus (called *designs*) in order to perform the logical explorations of (other) agents' knowledge. And such explorations will always have a definite logical meaning.

We may note that, with respect to our idea of interaction between Semantic Web agents as the search for handholds to establish *cuts* between logical theories, the first group of queries presents a situation where some robust handhold is already available, since the whole common language allows to pass from the knowledge of an agent (that is from its ontology) to the other's one, and the information (facts about individuals, or rough data) to flow between their A-boxes. On the contrary, the second group of queries acts on the basis of "loose" handholds, that is the elementary bricks of the standard (meta)languages, and the purpose of the queries is precisely to discover more interesting and meaningful *cuts*.

We admit that our presentation of such a desired lightweight query protocol is vague, somewhat impalpable, but we also consider that our intention is not to provide the protocol, rather to signal the need for it, the possible benefit that it would bring and also to step forward to show, with more details, how this form of interaction based on a query-answer mechanism can be profitably interpreted, understood and modeled by means of Ludics. Indeed, our concern in the whole course of this thesis is with the logical interpretation (the only reasonable sense of "meaning" to machines) of the information in the Web. And Ludics, as we are going to show, turns out to be really promising to deal with interactive processes wherein the agents exchange pieces of information which are easily identifiable with *addresses* (pointing to some namespace), and the use of that pieces is substantially comparable to *cut* discovery and *reduction* – which are among the main ingrediets of Ludics.

6.2. Ludics in brief

For the presentation of Ludics we largely rely on Fouqueré ([Fouqueré, 2011]) and Faggian ([Faggian, 2002]). Faggian clearly expresses the fact that "Ludics is to overcome the distinction between *syntax* (the formalism) and *semantics* (its interpretation) [. . .] Syntax and semantics meet in the notion of *design*" [Faggian, 2002]. The idea of a reconciliation between syntax and semantics, we believe, can be illuminating also with respect to the Semantic Web, where the discourse gets furtherly complicated by the ambiguity of the term semantics, which often leads people to think of the semantics of Semantic Web as something relevant to natural language meaning, but really it is just a restrictive interpretation, and we are convinced that it also limits and prevents the proper expression of the potentialities of Semantic Web. To put it in a nutshell, since the semantics of metadata is just another level of machine readable languages, layered over the syntax of other data, and it is to be handled syntactically as all other data, it has no point from a logical perspective to consider that (i.e. the metadata layer) as the semantics of the lower layer. In Logic, where Ludics produces the unification of syntax and semantics, this is possible through two opposite processes of concretization and abstraction. Indeed, syntax is to be made more abstract and semantics more concrete in order to have both going toward each other. Faggian identifies three key elements in these approaching processes:

Paraproofs. In the process of concretization (from semantics to syntax), the introduction of *paraproofs* allows to enlarge "the universe of proofs, in order to have enough inhabitants to be able to distinguish between them *inside* the system" [Faggian, 2002];

Focalization. In the opposite process, abstraction (from syntax to semantics), *focalization* ([Andreoli, 1992]) appears as an extremely useful instrument for the search of proofs. Based on previous results about the polarity of connectives in Linear Logic[5], it allows to group a series of consecutive actions (in a sequent calculus proof) of the same polarity as if they were a single logical action within the proof;

Locations. Still in the process of abstraction from syntax, *locations* mark the abandon of the formulas as objects to be manipulated within a proof. They are replaced by their *addresses*, the locations (or *loci*, plural of the Latin word *locus* for place) where formulas appear.

Together, focalization and locations allow to unveil the geometrical nature that underlies any logical proof constructed according to the technique of sequent calculus: one may forget the connectives and just look at an alternance of action of opposite polarity.

What is the interest of all this with respect to Semantic Web? It is worth slightly anticipating on the next section (that will be devoted to explain how Ludics applies to Semantic Web agents' interaction) in order to immediately give the flavour of our intuition. First of all, paraproofs allow for incomplete, partial and "wrong" proofs. This means in particular that we can obtain part of a proof as the result of the interaction of two agents, and such a part could be *per se* not a correct proof. But thanks to another interaction involving one of these agents with yet another one, the

[5]Basically, the logical connectives of LL can be distinguished in positive and negative – a division independent from that between multiplicative and additive connectives – the negative producing reversible actions within a proof, and the positive producing irreversible actions. Thus, $\oplus$ and $\otimes$ are positive whereas $\parr$ and $\&$ are negative.

proof resulting from this second exchange, though possibly once again incorrect alone, may compose with the previous proof a larger, correct proof. This makes room for two interesting possibilities to be taken into account in order to describe the activity of many ineracting, and collaborating, agents: i) the case where a single agent gathers pieces of information from different other agents, in a series of exchanges that finally allows to compose a correct proof (that is to achieve the task for which the agent had been activated); ii) and the case where many agents co-operate to compose a single correct proof by putting together pieces of knowledge / information that alone could be incorrect proofs.

Focalization, then, is that which enables the alternating dynamics of positive and negative actions. It is used in Ludics to represent proofs (more precisely the cut-reduction over proofs) as interactive processes between two players, where each player plays an action and then waits for the other agent's reply. From the point of view of Semantic Web agents – as we will describe better in a while – an interaction will look like a series of positive actions (those which the agent itself plays by choosing what to do) alternated with negative actions (where the agent just receives and acknowledges the replies from the other agent).

To say what locations stand for with respect to Semantic Web requires careful handling. Indeed, though Faggian ([Faggian, 2002]) counts the introduction of locations as an element of the strategy of abstraction from the syntax, we can see the role of such an element also on the part of a concretization of semantics. Indeed, if by Curry-Howard a formula can be identified with a type, we have that formulas as types in Semantic Web are tags and in particular the concept-names of ontologies. Precisely to the ambiguous nature of such concepts is attached most of Semantic Web confusion about semantics, and we would rather consider locations as that which helps in detaching types (as something with a logical meaning) from the "air of mystery" that sorrounds the notion of concept. Once made this point clear, the addresses, or *loci*, handled in Semantic Web agents' interactions can easily be seen as perfectly realized in the URIs that point to resources in the World Wide Web. Both concepts and punctual resources, all are actually syntactical objects (symbolic by their nature) appearing as URI, and even located within some namespace.

Now, to get into the technique of Ludics, we will present quickly its main ingredients: *designs*, the very objects of Ludics; the notion of *cut* between designs, that which activates communication; and *behaviours*, that which signals what a type really is in the syntactical semantics of Ludics. Indeed, we remark and insist on it, that the collapse of semantics and syntax in a single logically meaningful discourse gives us a semantics that does not attempt to escape from the realm of representations understandable to machines. We will find then that the communication between Semantic Web agents is nothing but cut reduction – as we finally have found that all operations on ontologies can be understood as *cuts*.

A *design* is an abstraction of a focalized proof in sequent calculus. In order to see more precisely what is the alternance between positive and negative actions in such a proof, we should look for a moment into a sequent and a sequent calculus proof. Very quickly, a sequent is a collection of logical formulas, that are progressively analysed and decomposed (simplified) into their subformulas in a process that is typically represented as a tree, with the root at the bottom which is the formula to be proved, also called *base*, and at the end of decomposing branches, as leaves, some axioms may be given. Sequent calculus, thus, provides the proof of a formula be reducing it to the truth of its premises and showing how truth is preserved through the series of logical operations that brings up to the final formula. A focalized proof, as suggested above, somewhat compresses the resulting tree by grouping all actions (that is application of a logical rule of sequent calculus) of a same polarity that happens before an action of the opposite polarity. Every action is actually a decomposition step of the formula that is chosen as the *focus* of the action for that "round". Since the main connective of a negative formula is a negative one, the "policy" with polarized proofs requires the formula that contains it to be decomposed immediately – it is the most convenient choice due to reversibility of the action. All subsequent negative decomposition steps are counted within the same step, so that next step will require a positive action. In case of a positive action, on the contrary, the focus may be chosen among the formulas produced by the previous action. Briefly, as [Fouqueré, 2011] explains, a decomposition step – corresponding to the application of a series of logical rules of same polarity in a sequent calculus – consists in

- either choosing a positive formula and decomposing it into a set of negative subformulas, thus producing a set of sequents (one for each negative formula),
- or decomposing the negative formula into a set of sets of (positive) subformulas, thus producing a set of sequents (one for each set of positive subformulas).

Now, to come back to Ludics, we have to substitute formulas with their addresses. As a consequence the decomposition path of a formula by its subformulas gets replaced by the "subaddressing" of an address, that is the progressive addition of a suffix (called *bias*) to every address that is "accessed" at each action.

Every *action* in Ludics then is an abstraction of some decomposition step. It has a polarity (positive or negative) and is noted by one focus, i.e. the address on which the action is focused, together with a finite set of biases, called a *ramification*. A special (only positive) action is the *daimon*, noted ✠. The base of a design is a set of addresses noted $\Upsilon \vdash \Lambda$, where Υ is either empty (and the base is positive) or a unique address (and the base is negative) and Λ is a finite set of addresses. A *chronicle* is a sequence of actions with distinct focuses that can be read on a design. In a chronicle positive and negative actions alternate, and the first action depends on the polarity of the base:

- if the base is positive, the first action is positive too and it can be either a normal positive action (on a focus plus some biases) or the daimon;
- if the base is negative, the first action will be negative too.

In particular, if a daimon appears in a chronicle, it concludes it. The focus of every action (which is not a daimon) is an address produced by one of the previous actions or it is present in the base of the design. The focus of a negative action is either in the base (and then the action is the first in the sequence) or it is produced by the action just before in the sequence.

A design on a basis $\Upsilon \vdash \Gamma$ then is a set of chronicles such that

- the set is prefix closed (chronicles start at the base and develop – even infinitely – by adding suffixes to the addresses);
- the nodes where the design branches are positive actions;
- actions occurring just after a branching with distinct foci for the different branches, must have distinct foci for all the rest of the sequence of actions (this allows to split chronicles);
- the leaves of the tree thus generated are positive actions;
- whenever the base is positive, the set of chronicles on it cannot be empty.

There are two main ways to represent designs: the first one recalls quite closely the trees of sequent calculus, focusing on the decomposition steps (the subaddressing of loci); the second one takes better care of the intuition of designs as sets of chronicles and draws a design as an arborescence, focusing on the actions (and somehow hiding the heavy accumulation of suffixes). For our examples we will prefer this last.

Cut-reduction (cut elimination, or normalization) – that which we have promised to reduce Semantic Web agents' interaction to – is the process occurring within special nets of designs known as *cut-nets* ([Girard, 2001]). A cut-net is made of a set of designs satisfying the following conditions concerning their bases:

- the loci in the bases must be pairwise disjoint or equal;
- loci which are equal must occur once in the left part of a base and once in the right part of another base. Thus they form a cut;
- the net resulting from cutting the bases must be acyclic and connected;
- one main design can be isolated, whose base either has the left part that does not get cut with any other base, or it has no left part.

We can now sketch how cut reduction proceeds between designs in a cut-net. Let $\mathfrak{D}$ be the main design in a cut-net, with first action (σ, I) or daimon:

- if the first action is the daimon, then the result of cut reduction is the design reduced to the daimon;
- if the first action is (σ, I) and the focus is a locus of $\mathfrak{D}$ that is not part of a cut, a commutation occurs and the process is relaunched for the subdesigns of $\mathfrak{D}$ accessible from that action;
- otherwise $\mathfrak{D}$ has no left part and its focus σ is part of a cut with another design with last rule $(\sigma, \mathcal{N})$ – where the *directory* (N) collects the ramifications of the actions on the same focus σ. In this case:
  - if $I \notin \mathcal{N}$, then interaction fails;
  - otherwise, the reduction continues on the connected parts of the subdesigns obtained considering only the part I of $\mathcal{N}$.

Therefore, cut reduction either fails, or it never ends, or it ends up with a design that reduces to ✠. In this last case the cut net is said to be closed and no commutation step occurs during the reduction. When the cut reduction between two designs reduces to ✠, those designs are said to be orthogonal (to each other) – and can be noted $\mathfrak{D}$ and $\mathfrak{D}^{\perp}$.

The notion of *behaviour* is built on that of orthogonal designs: a behaviour is a set of designs equal to its bi-orthogonal. That is, all its designs successfully normalize with all the same counter-designs (i.e. orthogonal designs). It is important to signal that cut reduction is usually explicitly considered, within the frame of Ludics, as the natural form of *interaction* proper to logical proofs. Indeed, thanks to the enlargement of the universe of logical proofs by taking into account also "wrong" proofs (the paraproofs), the cut reduction appears now as the process by means of which the correct proofs can be identified as those leading to successful normalization. Actually, since a cut reduction, as an interaction between (para)proofs, involves at least two designs it is not yet possible to decide which one represent the correct proof (the one that Frege and the tradition of truth-valued semantics would call the proof of a truth). The criterion, wholly internal to Ludics, is that the "winning" design does not play the daimon during the interaction[6], so that this allows finally to identify correct proofs. But then, what is that gets proved by a design? Behaviours are the answer, as far as they denote formulas. Indeed, a set of designs equal to its bi-orthogonal contains all the correct proofs that prove a same formula – since they lead to successful interaction (cut reduction) with the same set of counter-designs.

Moreover, like a chronicle "walks" a linear *path* over a design, from the base up to some leaf, possibly a daimon, enumerating the actions of a sequence, so we have an interactive correspondent of the chronicle in the *dispute*. Besides its formal definition (for which we simply remand to [Girard, 2001] since it would require too deep an immersion in technical matter), a *dispute* can be seen as the matching of two paths over two orthogonal designs. In particular, the dispute appears as a travel on the different chronicles, where actions in a chronicle are matched by counter-actions in the other chronicle, and the dispute stitches the chronicles together by jumping from one to the other on negative actions (see figure 6.1 for an immediate explanation), up to the moment when one of the two chronicles introduces the daimon, thus stopping the interaction – and indicating the other chronicle's design as the real proof. Let us observe figure 6.1, that we reuse by courtesy of Fouqueré ([Fouqueré, 2011]). We recall that of the two ways of representing Ludics designs we adopt the arborescent style focusing more on actions than on decomposition steps (subaddressing). Moreover, here we lighten it yet more by hiding ramifications, so that for each action we can see just its focus and polarity – after all it is enough to give the essential ideas that we are going to reuse for Semantic Web interaction. Now we know that each tree in the figure is the abstraction of some sequent calculus proof, in particular the rightmost tree is the abstraction of the normalized proof obtained by cutting together the two proofs represented by the designs on the left. Nodes in the trees signal actions (possibly summing up a series of logical rules of the calculus which have the same polarity, as in

[6] Typically at the end since it stops the interaction.

polarized calculus) and circled actions are the positive ones. If a chronicle is any walk of a path (over a single design) from the root to some leaf, the dispute is shown in the figure as the travel on a path that stripes over the two designs on the left, that finally leads to the reduced (normalized) tree on the right, where cut nodes have disappeared. Indeed, whereas dashed arrows signal commutations in the cut elimination process (and look as moves within a single design), solid arrows signal the cuts and connect a positive action of a design with the negative (corresponding) action in the other design – so that jumps are always on negative actions. Obviously, the cut may take place wherever two designs present two corresponding actions (with same focus and one matching ramification) of opposite polarity.

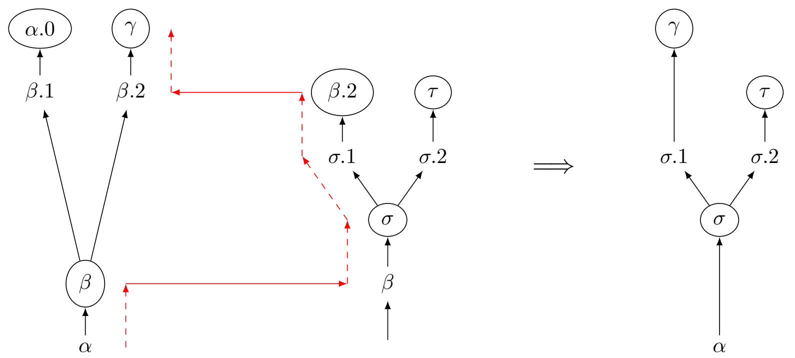


Figure 6.1.: Cut elimination between two designs

6.3. Ludic querying

Now that we have introduced both the processes that properly should characterize Semantic Web agents' interaction and the main elements of Ludics, we can attempt to establish some correspondences between these two worlds in such a way to exploit Ludics as a theoretical framework to account for Semantic Web agent's communication. We must recall, however, that this part of the work is a pretty recent achievement of our research and we, therefore, propose it as a work in progress, with many points that surely would deserve deeper consideration and, perhaps, also partial rethinking to be brought to their optimal setting. The fundamental correspondences that we are able to settle at present are:

- a chronicle is any possible exploration of an agent's Knowledge Base (KB);
- a design then, as a set of chronicles, is a representation of the knowledge of an agent – as to say: the knowledge of an agent is represented by the set of all the "discourses" that it is able to tell;

- interaction happens between (at least) two agents that engage in a query-answer dialogue;
- networks of designs (cut nets) may represent the interaction between many agents (in Semantic Web too there is no reason to constrain the interaction to only two agents at a time).

Interaction then, i.e. the logical process of cut reduction, takes here openly the form of *communication*, where the cut between (the bases of) two designs – represented by a same locus in opposite positions in the base sequents – is precisely the matching between a query and its answer, the interest of an agent asking about something and the ability of another agent to give information (reply) about that. And cut reduction can be interpreted in the query expansion, that allows to reach possibly for "terminal information" such as data, A-box individuals, general Web resources. Concerning the mechanism of the dialogue, we find in particular that:

- from the viewpoint of the interrogating agent, the positive actions of a dispute are the daimon (✠) – meaning that it is satisfied and has nothing more to ask – or some query, whereas negative actions are expectations of answer on the part of the other agent;
- from the viewpoint of the answering agent, the positive actions of a dispute are the daimon (✠) – meaning here that it is not able to provide information – or its answers to some query, whereas negative actions are expectations of query on the part of the other agent;
- a dispute is the matching of a chronicle (the exploration of the answering agent's KB) against the series of queries posed by the interrogating agent;
- an interaction succeeds when it ends by a daimon played by the interrogating agent;
- otherwise the interaction fails when it ends by a daimon played by the answering agent.

We may exclude here the case of never ending interactions due to the inevitably limited dimension of an agent's knowledge. With more details we may note also that in a design:

- *loci* are addresses of some namespace;
- an action is made of (and uniquely represented by) the locus that is its focus, plus a ramification, that is the set of (sets of) biases that are to mark subaddresses of the focus in subsequent actions;
- both queries and answers can be considered as interaction requests (cf. [Fouqueré, 2011]) and all lead to some new locus;

- at each positive action on the part of the interrogating agent, either a daimon is played, or a locus is promoted as *focus* of the action by being joined with some query construct (possibly out of the set of standard queries that we have suggested above talking about a desired lightweight protocol for agents' interaction)[7]. The query chosen provides the bias element that signs the action together with the focus chosen – ramifications here are just singletons;
- at each positive action on the part of the answering agent, either a daimon is played, or the focus is automatically given (this agent has to reply to the query) and the action is signed with the ramifications containing all the addresses (loci) that are correct answers to the query.

It needs here a short parenthesis concerning the addresses of such interaction designs. These addresses indeed can be either concept-names (terms from the vocabulary of some ontology) or the names of individuals appearing in the KB at some time (or even data values relevant to these last). Hence, in spite of the plain suffixing (i.e. addition of suffixes to the loci used) that is into play with standard Ludics designs, we have that other (new) subaddresses appear, even placed within different namespaces with respect to the namespace of the focus. Thus, still remaining a purely syntactical process, subaddressing by the cut reduction corresponds in principle to the progressive specification of the meaning of some term thanks to the specifications formalized and stored in ontologies and KBs. After all, such kind of interaction, as we said, is *communication* aiming at the share of knowledge.

We may consider that an interaction starts with a positive action by an interrogating agent played toward another agent; this first action could be the standard "shakehands" query that asks the other agent about the vocabularies that it knows – so that we may also consider that every agent not committed in performing some task stays in a sleeping mode, waiting for such a kind of query. The following action of the answering agent will then provide the interrogating agent with the list of known namespaces (given as the ramifications of that action). The interrogating agent accepts the answer, and chooses from that list the namespace to query – that is promote a locus as the focus of its next action – and fires another query, that could be now something like the request for all the terms that the other agent knows from a selected namespace. The answering agent, in its turn, accepts the query and must operate, again, on the unique ramification provided by the interrogating agent. The interaction goes forth this way up to the moment when one of the agents plays the daimon. If it is played by the interrogating agent, then the interaction has succeded, otherwise it has failed. That is in particular, the answering agent has no information to send back to the interrogating agent. The success of the communication (as a logical cut reduction in terms of Ludics) is the possibility to prove an information that is needed (it is asked by means of a query indeed) by the interrogating agent. This last "looses" in the interaction just in the sense that it is not it that proves the required information, rather it is the other agent that succeeds in that, so that it has in its knowledge a "winning

[7]With respect to the production of positive actions described in [Fouqueré, 2011], such query construct could be thought of as links being always available for every locus.

discourse" (or strategy) to prove that information.

Figure 6.2 shows quite clearly the basic mechanism that we consider at the basis of Semantic Web agents' communication.

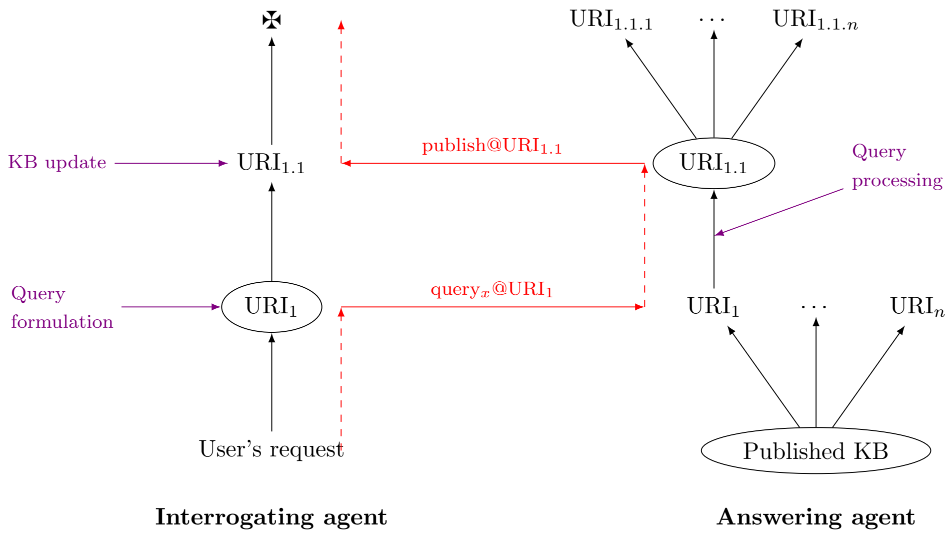


Figure 6.2.: Basic example of communication between Semantic Web agents

Let us remark some points by observing the figure. We observe that:

- the design on the left represents the knowledge of the interrogating agent and how it gets expanded thanks to the interaction with the other agent;
- the design on the right represents the knowledge of the answering agent. In particular, every walk from the root to a leaf represents a "discourse" that the agent is able to talk, thus proving some information (or fact);
- the interrogating agent is activated by its master's request to accomplish some task (typically looking for some information in the World Wide Web). Thus, we signal the initial status of the interrogating agent as a negative one;
- the answering agent is in a proactive waiting state, that is, it is publishing on the Web its knowledge base, so that any other agent may pose queries on that;
- the cut links between designs signal the match between a request for communication about some subject (the focus of the action) on the part of one agent, and the disposition to interact on that same subject on the part of the other agent. In particular:

  - in case of a cut jumping on the answering agent's side, the request for interaction is a query, and the cut can take place because the answering agent "knows" the term (an URI) that is the focus of the query;
  - in case of a cut jumping on the interrogating agent's side, the request for interaction appears as an answer to some query and the cut brings new information in the interrogating agent's KB, that can be updated.

- after a cut has occurred, a commutation step
  - on the answering agent's design signals the processing of the request contained in the query and leads to the output of that query over its KB. The output may be either the provision to the other agent of a piece of information from its KB, or the impossibility to give an answer, signalled by the daimon;
  - on the interrogating agent's design signals the updating of the KB, that may lead either to another query to pose or (if the update has provided the information required by the user's request) to the conclusion of the interaction (signalled by the daimon).
  - in any case, when the dispute (the travel jumping between paths of the different designs) passes on a daimon, then interaction stops and communication ends.

Additionally we remark also that:

- the answering agent accepts any query focusing on a URI (a term) that appears in its KB – this allows to establish a cut. Then, interaction may fail if either the query construct to be expanded in order to answer the query is not known by the answering agent (that is, the query_x in the figure is not a query construct of the lightweight query protocol that we have imagined), or its KB does not allow to give an answer to the query[8];

- since the *loci* in this kind of designs are URIs their appearence in positive and negative forms depends on just the attitude of the agent about the term (asking about it or knowing it);

- that which looks like subaddressing in the figure (URI_1, $\text{URI}_{1.1}$ and so on) is not real subaddressing – since we have not subterms in namespaces – rather it is just to signal the terms accessible from that term in the (name)*space* of the agent's KB.

Finally, in considering how this kind of communication between Semantic Web agents could develop, we must consider that an agent like these has no own intentionality. Therefore, communication as exploration of other agents' KBs is to be conceived, in first approximation, as an exploration as both extensive and deep as possible, by trials and errors.

[8] It is about the same thing as one replying "I don't know" to a question.

6.3.1. Deeper interpretation

Such a dynamics could be hastily seen as just a somewhat complicated game semantics for agents' interaction. But to interpret Semantic Web agents' interaction by means of Ludics is pretty more instructive than that, since it allows for a deeper interpretation of the interaction that, we believe, matches marvellously with the Web environment. Actually, if one looks to Ludics as a game semantics, the victory does not depend on the winner, but on the abandon of the game on the part of the other player, the "loser". And the defeat really is the abandon of the match, that is the stop of the collaborative exchange of information, as in a dialogue. Briefly, there is no competition, rather collaboration, cooperation; it matches the setting of World Wide Web, where the stake of the game is information, which is not to be catched, but shared: as a game it is not a zero sum, but a positive sum game.

Another very interesting aspect of this proposal is the one that connects Ludics to ontologies, and in particular to concept discovery – the detection of concepts as emerging from interaction, that is from communication. Indeed, we can see that the meaning of a term is actually discovered at the end of a successful interaction. At the beginning of every dialogue (query-answer interaction) a concept (that is a type) is known only by the term (the address pointing to some namespace) that is chosen to be used as a label on the metadata tags that apply it (the concept) to web resources. But not only that corresponding (human mind's) concept is actually unknown, but also its logical interpretation (of the term) may be unknown – it is the common case of a concept-name appearing in an A-box which can be understood only by matching it with the relevant T-box. Therefore, the representation of the query-answer process by means of Ludics is to offer also an interesting perspective on the meaning of any queried term and, as a consequence, of the corresponding concept. Indeed, based on the notion of *behaviour* proper to Ludics one can recover an approximation of the concept expressed by the queried term by the closure under bi-orthogonal. In other words, with respect to the account of Semantic Web agents' interaction that we have just sketched, this means that everything that would receive the same answer during an interaction forms a concept. Moreover, it can be equated with all the other terms that would get the same answers.

We can see this through both the kinds of queries possible in our ideal lightweight querying protocol, that corresponds after all to the two traditional perspectives available on a Knowledge Base. After an interaction process that involves only T-box exploration queries, we have that any queried term gets defined based on the logical definition(s) that can be found in the answering agent's KB. If there are two terms that get the same definition as answer, then the two terms express the same concept within that particular interaction, between those agents involved and as far as their KB are not updated. This is a very important remark: the bi-orthogonal closure indeed holds for each particular query answer session; the closure of a Ludics behaviour and therefore the meaning of the corresponding concept of some ontology is scoped to the agents and KBs involved in each particular interaction. But this is not different from any other kind of interaction happening on the Web: results of a query asked to a server change over time and also based on the agent or user that poses the query. Moreover, as we will discuss more deeply in the last part of this work, the same can be true even for communication between human beings. We can also see the same principle holding when looking from the extensional perspective, by means of data retrieval queries: any two terms that could receive the same set of individuals (data, resources and the like) as answer, during a given query-answer session, will be deemed to be the same concept. This is logically compelling and also correct for all the same reasons that we have just listed, provided that the discourse is scoped to the particular query-answer session, to the particular agents involved and to their particular KBs. By the way, we find such a result perfectly matching with our previous effort to render ontologies as Ontological Compatibility Spaces: an OCS perfectly defines a notion of compatibility that is scoped to the single KB dealt with.

After all, this is quite natural in a query-answer process. Any query indeed builds up some new concept or relation of an ontology – the same holds for databases too, where the basic structure of tables expresses relations and every query takes a *view* of that relation. In the simplest case the query asks only to recover data corresponding to the concept/view mentioned in the query itself – to our setting, it is the case of instance (or resource) retrieval based on the tags they bear. Such a simple case is already enough to have the principle of bi-closure come into play: any other query that will obtain the same answer should be considered equivalent to the first one – provided that one limits the discourse to the portion of world that the system may access, that is the (set of) data source(s) on which the queries are executed.

But more interesting forms of query will ask the definition of a term. It is the above-mentioned case where an agent "follows" the namespace track from an A-box to a T-box to recover the definition of a concept name. It is likely to get as answer a (more or less logically strict) definition, based on which it can go further in asking for the terms that appear in that definition, and so on. In this way, the notion of *dispute* proper to Ludics fits well also our representation of query-answer processes, since it describes the exploration of the knowledge of an agent (the one who "owns" the T-box) by the other agent. Here again, the closure principle provides us with the determination of the intended concept as the set of terms that could be expanded in the same way – and possibly even get, at the top of the design – the same set of data items (resources, individuals).

To conclude the argument about the reliability of this mechanism for "concept discovery" we may call for an "ally" above any suspicion – or better, two. Genesereth and Ketchpel, finally gave in [Genesereth and Ketchpel, 1994] (in 1994) the definition of agent that we have quoted above (at p. 180). We recall here just the last part: the agent "must abide by the behavioral constraints implicit in the meanings of those messages", that is the messages that it exchanges with other agents in their specific language (ACL according to Genesereth and Ketchpel, or our lightweight query protocol accoridng to us in the case of Semantic Web). And we consider that the best identifiable behavioural constraints in the case of Semantic Web interactions have primarily the form of the information that comes back from other agents: first of all, the answers that an agent receives constrain its possibilities for subsequent actions; and, lastly, the data (or web resources) that an agent may find as satisfying the research that it initiated (probably in order to accomplish a task on behalf of some human user) *are* the final constraint that binds definitely that interaction to something that satisfies its research and can be used to accomplish its task. The meaning, therefore, *is* those constraints.

6.3.2. Example – Ludic-queries

Besides the illustration of the general case of interaction between Semantic Web agents (cf. figure 6.2) we propose here a couple of more detailed examples about the simple KB that we have described in chapter 2 and used for other examples in chapter 5. It is the usual FOAF KB, describing two persons together with their email addresses and the web page of one of them. The idea with these examples is just to suggest a little better what could look like a querying session between Semantic Web agents. We may represent both the examples on the same figure 6.3.2, so as to suggest also that every particular (and unique) "dialogue" between two agents is one particular dispute, that is one travel across the involved designs, among many alternative possible travels. Indeed the two examples are structured as two query sessions between the same agents. In particular it is noteworthy that the design on the right, where the actions of the answering agent are shown, is the same in both examples, but gets explored along different paths. The idea lying behind this is that an answering agent simply publishes, makes available its knowledge and the particular path that will be explored in a given dialogue depends on the querying agent, according to the queries it will ask. On the left of the figure we have the two trees (albeit unilinear) corresponding to two different strategies on the part of the querying agent. The second one is represented as an "engraftation" onto the first one (which is the one ending first by the lower daimon) because in this way we may present the two dialogues as occurring one after the other, one as the prosecution of the other. Indeed, for the second dialogue the interrogating agent uses knowledge and information that it has just acquired thanks to the previous query session – we recall that according to our idea of interaction between web agents these last learn, that is acquire new "pieces" of knowledge that get sewn to their original KBs, enlarging them time by time. On the other hand, one could also consider these two designs / strategies as being really just one strategy that gets exploited in two alternative ways, allowing for different disputes between the same (only two) designs, but yet producing two distinct dialogues with the other agent[9]. (Note that this last interpretation would require to erase the edge that links the `User's request` node to the `q4` node). Along with its arborescent representation we give below also a textual interpretation of these examples of interaction. We proceed by explaining the tags on nodes first of all:

User's request as usual is that which activate the interrogating agent by assigning it some task to perform (some information to find in our case);

Published KB on the other hand signals the availability of the answering agent to begin a query session – we could see that as the status of every agent before a

[9]Concerning such a splitting of designs, we could even speak of *delocation* (for a complete account of Ludics designs and their working we refer to their inventor Girard [Girard, 2001]). For the time being however, given the early, and a bit immature, stage of this proposal for a Ludics style interpretation of web agents' interaction, we just suggest it as a margin note.

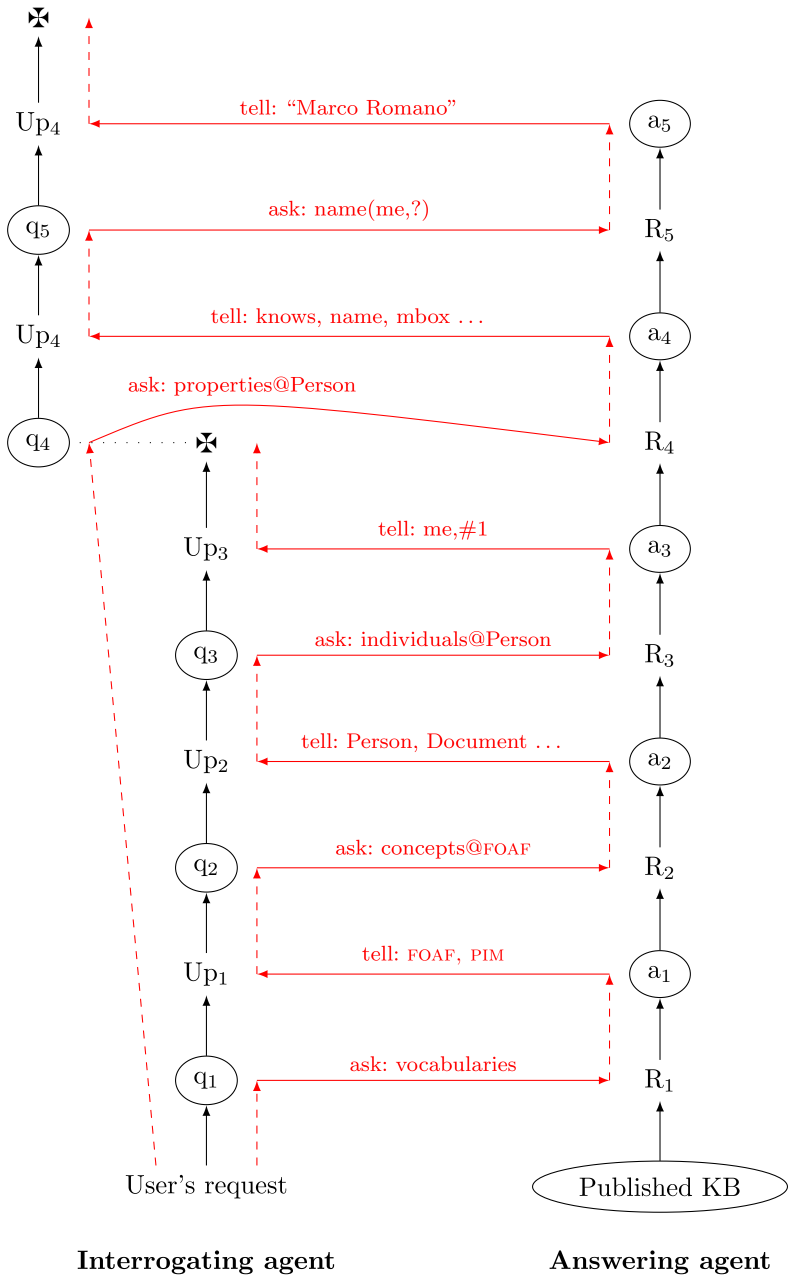


Figure 6.3.: Representation of two example interactions between Semantic Web agents

user assigns it some task;

q_n marks any further positive action (up to the daimon) on the part of the interrogating agent. It signals the formulation of some query by applying some query construct (or query operator) provided by the (imagined) lightweight query protocol and choosing a particular *locus* as *focus* of that action, among the *loci* available at that stage of the interaction in the interrogating agent's KB. It can be either a term (tag, concept-name, individual name, in general URIs) that was already in the agent's KB before the dialogue began, or a new term it has just learnt thanks to the previous steps of the interaction;

R_n marks the negative actions on the part of the answering agents. It signals the successful receipt of the query, as the focus of the query matches some term in the answering agent's KB;

a_n marks every further positive action (up to a possible daimon) on the part of the answering agent. It signals the reply to the interrogating agent's query, which is sent the requested information;

Up_n marks the negative actions on the part of the interrogating agent. It signals the updating of this agent's knowledge base with new information coming from the other agent.

Let us consider now the solid arrows that actually produce the dialogue, i.e. the *cuts*.

1. The first query (that we have imaginatively noted `ask: vocabularies` in the wait for a real query protocol) is a sort of "shakehands" step by which the interrogating agent discovers which vocabularies (ontologies) the other agent knows, so that it is possible to determine a common language. The cut occurs since the answering agent accepts the incoming query (basically, it knows the minimal language in which it is expressed);

2. the second cut then (labelled by `tell:` FOAF, PIM), is the transfer (copy) of information from the answering to the interrogating agent. The cut takes place since the interrogating agent too knows the (minimal) language – that we may easily suppose defined by the query protocol itself – with which the new information (terms, URIs) coming from the answering agent is "wrapped";

3. next cut is given by the query `ask: concepts@`FOAF that is a request for the list of T-box terms (concept-names) available in the FOAF vocabulary. One may note that here the interrogating agent chooses as focus of its new request something that it has just "learnt";

4. the subsequent cut labelled by `tell: Person, Document ...` is the answer by which the other agent replies to the query, listing all the concept-names that it knows as pertaining to the namespace of the FOAF vocabulary;

5. by posing the query `ask: individuals@Person` the interrogating agent asks for a list of individuals that instantiating the concept `Person`. Please note that the scope of such a query is always limited to the other agent's KB

6. which in turns replies by listing all the individuals that it knows have the type FOAF:`Person`. In this case the answering agent knows two individuals: the one named `me` – which is likely to correspond to the "owner" of the answering agent – and a blank node (#1) which is an entity described within the KB, but which is not given an explicit URI (cf. the A-box on page 52). To shed some light on this "mystery" we recall that the example A-box describes this author (`me`) and tells about his acquaintance with someone named V. Michele Abrusci – actually, to be precise, the recorded information within the A-box sounds like *something* named like that, and this is the reason for the blank node[10];

The first dialogue ends here, with the presentation of all the (FOAF) persons that the answering agent knows, thus contributing to the "understanding" of this concept (precisely FOAF:`Person`) by the interrogating agent with the presentation of those two individuals that it knows as instances of it. The second dialogue goes after this point:

1. the interrogating agent asks `ask: properties@Person` that is the request for the list of properties – both roles and actual properties, known in OWL as object properties and datatype properties respectively – that holds for the concept `Person` in FOAF as it knows this concept by the previous exchanges;

2. the answer (`tell: knows, name, mbox ...`) then provides this list of properties. By the way, whereas FOAF:`knows` is a role, i.e. relates individuals (in this case instances of `Person`), FOAF:`name` is a datatype property, whose values are dealt with by all Semantic Web applications as mere data values;

3. last request by the interrogating agent (`ask: name(me,?)`) asks for the `name` of the individual previously discovered, known as `me`;

4. the answering agent then has just to retrieve the value and send it back: `‘‘Marco Romano’’`.

In both dialogues, the interaction is stopped by the interrogating agent that plays the daimon (✠), that could be rephrased as just "Thanks, that's all".

6.4. Open issues

We could say that this chapter is the edge of our research. We began this thesis with the objective to give some contribution for a better understanding of interaction in the Web, especially insofar as this last gets organized as an extremely rich source of information and knowledge understandable to machines. Therefore, our interest was

[10] We just recall in addition that in the general case, a "run of reasoner" could provide additional results for queries like that. But we do not assume the use of a reasoner here.

to signal the need for a deeper consideration of web agents' interaction as something that is quite different from the usual client-server protocols. Even Semantic Web services indeed, by relying on that same approach, make a poor use of the knowledge spread in the World Wide Web. We really believe that, on the contrary, the kind of interaction that we have just attempted to describe could be a great improvement in the exploitation of the Web. But the proposal that we managed to present here in order to provide a theoretical framework for it, is apparently immature.

For instance, some aspects of Ludics still need a neat interpretation in the context of Semantic Web agents' interaction and, most of all, aspects of Semantic Web agents' interaction that we may have disregarded till now should obviously find, they too, the proper interpretation in terms of Ludics interaction. Moreover, the connection between Ontological Compatibility Spaces, as we have defined them in the previous chapter, with the use of Ludics that we have introduced in this chapter – though conceptually quite apparent – is yet to be formally specified. The timing of research may not coincide with that of academic years, but we hope to be able – and have the occasion – to continue such a research in the next future.

As a consequence, we list here the most important open issues that we can see at present about our proposal to support, by means of OCSs and Ludics, a more interesting, and useful, form of interaction between agents in the World Wide Web:

- the main issue concerns how to finely put together OCSs and Ludic-queries. It is even more thorny especially after that, in the previous chapter, we have depicted the orthogonal of an ontology (and with it, of all its concepts) by means of the notion of the dual OCSs. Now, the point is how to reconcile the anticliques of the dual OCS as, also, the place where one should interpret queries, that is the questions to which the cliques of a positive OCS are answers;

- another important point is to effectively describe interaction between more than two agents by means of a network of designs;

- and finally, various elements of Ludics (such as incarnation and delocation just to name some) that we have not considered should be analysed in search for other contributions from Ludics to Semantic Web.

Bibliography

Vito Michele Abrusci, Marco Romano, and Christophe Fouqueré. Knowledge Representation through Coherence Spaces. In Jan L.G. Dietz, editor, *proceedings of KEOD 2009, 1st International Conference on Knowledge Engineering and Ontology Developmnet*, pages 220–225. INSTICC - Institute for Systems and Technologies of Information, Control and Communication, 2009. ISBN 978-989-674-012-2.

Vito Michele Abrusci, Marco Romano, and Christophe Fouqueré. Ontologies and Coherence Spaces. In *Ludics, Dialogue and Interaction*, volume 6505 of *FoLLI LNAI*, pages 205–219. Springer-Verlag, 2011.

J.-M. Andreoli. Logic Programming with Focusing Proofs in Linear Logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.

Tim Berners-Lee, Dan Connolly, and alii. A PIM ontology. `http://www.w3.org/2000/10/swap/pim/contact`, 2000.

Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez Muro, and Riccardo Rosati. Techniques for Ontology Interoperation (Deliverable 24). Technical report, TONES – Thinking ONtologiES, EU-FP6 project, 2006.

C. C. Chang and H. J. Keisler. *Model Theory*. North-Holland, 1976.

Claudia Faggian. Travelling on Designs. In *Proceedings of the 16th International Workshop and 11th Annual Conference of the EACSL on Computer Science Logic*, CSL '02, pages 427–441, London, UK, 2002. Springer-Verlag.

Christophe Fouqueré. Ludics and Web: Another Reading of Standard Operations. In *Ludics, Dialogue and Interaction*, volume 6505 of *FoLLI LNAI*. Springer-Verlag, 2011.

Stan Franklin and Art Graesser. Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. In *Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages*, pages 21–35, London, UK, 1997. Springer-Verlag.

Bernhard Ganter, Gerd Stumme, and Rudolf Wille, editors. *Formal Concept Analysis: Foundations and Applications*, volume 3626 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2005.

Michael R. Genesereth and Steven P. Ketchpel. Software Agents. *Communications of the ACM*, 37:48–53, 1994.

Silvio Ghilardi, Carsten Lutz, and Frank Wolter. Did I Damage my Ontology? A Case for Conservative Extensions in Description Logics. In *In Proceedings of KR2006*, pages 187–197. AAAI Press, 2006.

Jean-Yves Girard. Linear Logic. *Theoretical Computer Science*, 50:1–102, 1987.

Jean-Yves Girard. Locus solum: from the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, pages 301–506, 2001.

Jean-Yves Girard. *Le Point Aveugle: Tome 1. Cours de Logique, Vers la perfection.* Hermann, July 2006.

Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types.* Cambridge University Press, 1989.

Bernardo Cuenca Grau, Carsten Lutz, Ulrike Sattler, and Anni-Yasmin Turhan. Tasks for Ontology Integration and Merging (Deliverable 11). Technical report, TONES – Thinking ONtologiES, EU-FP6 project, 2006.

Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. A logical framework for modular integration of ontologies. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 298–303, 2007.

Thomas Gruber. Ontology of Folksonomy: A Mash-up of Apples and Oranges. *International Journal On Semantic Web and Information Systems*, 3(1):1–11, 2007. available also online at `http://tomgruber.org/writing/ontology-of-folksonomy.htm`.

Tom Gruber. Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human and Computer Studies*, 43(5-6):907–928, November 1995.

Nicola Guarino. Formal Ontology and Information Systems. In *1st International Conference on Formal Ontologies in Information Systems, FOIS'98*, pages 3–15. IOS Press, Trento, Italy, 1998.

Nicola Guarino, Massimiliano Carrara, and Pierdaniele Giaretta. Formalizing ontological commitments. In *AAAI'94: Proceedings of the 12th National conference on Artificial Intelligence*, volume 1, pages 560–567. American Association for Artificial Intelligence, 1994.

P. Hayes. RDF Semantics. `http://www.w3.org/TR/2004/REC-rdf-mt-20040210/`, 2004.

Andreas Hotho, Robert Jaeschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, pages 411–426. Springer, 2006.

Robert Jaschke, Andreas Hotho, Christoph Schmitz, Bernhard Ganter, and Gerd Stumme. Discovering shared conceptualizations in folksonomies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6:38–53, 2008.

Yu-Kyung Kang, Suk-Hyung Hwang, and Kyoung-Mo Yang. FCA-based conceptual knowledge discovery in Folksonomy. In *World Academy of Science, Engineering and Technology*, volume 53, pages 842–846. 2009.

Alain Lecomte and Myriam Quatrini. Ludics and Its Applications to Natural Language Semantics. In *Logic, Language, Information and Computation. 16th International Workshop, WoLLIC 2009. Proceedings*, pages 242–255. Springer Berlin Heidelberg, 2009.

Maurizio Lenzerini. Data integration: A theoretical perspective. In *PODS'02: Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of database systems*, pages 233–246. ACM Press, New York, 2002.

Obitko Marek. `http://www.obitko.com/tutorials/ontologies-semantic-web/operations-on-ontologies.html`, 2006.

Peter McBrien and Alexandra Poulovassilis. P2p query reformulation over both-as-view data transformation rules. In Gianluca Moro, Sonia Bergamaschi, Sam Joseph, Jean-Henry Morin, and Aris Ouksel, editors, *Databases, Information Systems, and Peer-to-Peer Computing*, volume 4125 of *Lecture Notes in Computer Science*, pages 310–322. Springer Berlin / Heidelberg, 2007.

Peter Mika. Ontologies are us: A unified model of social networks and semantics. In *Web Semantics: Science, Services and Agents on the World Wide Web*, volume 5, pages 5–15. 2007.

Natalya Fridman Noy and Mark A. Musen. An algorithm for merging and aligning ontologies: Automation and tool support. In *In Proceedings of the Workshop on Ontology Management at the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*. AAAI Press, 1999.

Jeff Z. Pan, Stuart Taylor, and Edward Thomas. Reducing Ambiguity in Tagging Systems with Folksonomy Search Expansion. In *ESWC 2009 Heraklion: Proceedings of the 6th European Semantic Web Conference on The Semantic Web*, pages 669–683, Berlin, Heidelberg, 2009. Springer-Verlag.

C.I. Petrie. Agent-based engineering, the Web, and Intelligence. *IEEE Expert*, 11(6): 24–29, 1996.

Part III.

Philosophical issues

What philosophy may say about all this?

In this last part we finally concentrate on the most theoretical aspects that are involved with the use of ontologies as knowledge representations for the Semantic Web, attempting to provide a more philosophically aware interpretation of all the subject. We take multiple aims in doing this. First intention is to provide a comprehensive account of ontology capable to illustrate, on the one hand, how close is the relationship between formalized ontologies for information systems and current research in (formal) ontology, but also capable, on the other hand, to identify and put out original and peculiar problems of formalized ontologies for Semantic Web, which – as we will show – are not only technical matter. Rather, they pose interesting challenges to philophers who take seriously those issues, and deserve an adequate treatment. Obviously, we do not claim to have definitely solved any of them, but we may say to be happy to signal them and propose our way to face them.

Second aim, but tightly bound with the previous one, is the aim to raise philosophical interest on Web ontologies. Actually, there are already many philosophers involved with computational ontology (ontology for automated information systems), but as far as we know, they all deal with formal ontological research. That is, they are ontologists lent to applied ontology, who typically propose readjustmnets of their formal ontological theories for use in the Web or within restricted information systems. They carefully deal with formalization issues and scrupolously introduce their formal method to computational ontology, enriching knowledge engineers' communities with it. But nobody considers the other face of the coin: the use of these ontologies. It turns out to be, basically, a matter of communication. Indeed, to use ontologies calls for exchanges between ontologies; and (Web) ontologies are that which grounds an artificial language to some reality whose status and nature is at least cryptic. We therefore will try, in next chapter, to shed some light on this point in order to signal its relevance to philosophical enquiry and its central importance to the development perspectives of World Wide Web. We will also indicate a way to approach this issue that would surely take advantages by the development of just presented ideas concerning the use of Ludics to query ontologies (and other forms of knowledge representation / information description) represented as Ontological Compatibility Spaces.

Besides, in this section we aim also to bring to some conclusion many issues that we have raised in the previous parts of this work. This will take place in particular in the last chapter, together with our final, conlusive observations at the end of this work. It is precisely because such observations may be considered as the results on the philosophical side of this work. The main point, and *leitmotif*, will be the "discovery" of multiplicity in ontology. Obviously this will not be approached as a matter of freedom of expression for a plurality of subjects – it is not relevant to our work – but as the prerequisite to real interaction. It is just on the basis of a plurality of perspectives that comparison may take place and, therefore, a fruitful communication may lead to new discoveries, better understandings and, finally, richer definitions of what is there. We have tried, in the previous part of this work, to make it appear in the special setting of Semantic Web, in the interplay between autonomous, intelligent web agents that share knowledge and possibly learn something from each other, arriving at (new) shared definitions of the concepts in their ontologies. And here, finally, we try to suggest how and why the same dynamics can be fruitful in the wider field of (philosophical) ontology.

Concerning the structure of this last part of the work, it is made of just two chapters. The last one then is the one that we have just described: our conclusions, that will be introduced necessarily by means of an original understanding of what is going on in Semantic Web and, as a consequence, in ontological research.

The first chapter instead (next pages), will deal mostly with a "summary" of the history of ontology. Of course, it might look like quite pretentious to attempt, within a thesis about ontologies as knowledge representations, to put in a single chapter that which has been for millennia one of the fundamental problems of philosophy – if not the fundamental. And indeed, it would really be pretentious to do something like that. But we aim to do quite a different thing. Namely, we aim to identify throughout the history of ontology, a few crucial steps where either the purpose or the scope, and sometimes both together, take a turn, or a shift, that changes the method of performing ontological research and the expectations about the results of ontological research. We think that it is the key to both a full understanding of the value of nowadays computational ontologies and also of their philosophical relevance.

In pursuing this objective within a thesis that aspires to offer some interesting result both on the side of the logical structuring of information for the Web and on the side of a more general, overall philosophical regard to ontology – not to say about the interest for the communication dynamics – we could not operate, and somehow apologize for that, that typical part of the research work of a philosopher that consists in attaining the primary sources. In fact, as far as we deal with ancient philosphers, and up to modern authors, we in general rely on secondary literature. When we sometimes refer to primary literature, if not otherwise specified, we however rely on English translations (in any case at least reliable if not authoritative) made publicly available in the World Wide Web within projects aimed to spread philosophical classics and improve their accessibility. After all, by presenting such a thesis we cannot but trust the Web as the place of shared knowledge.

Moreover, also the scarcity of quotations and references to others' works, indeed an oddity if not a fault in a philosophical work, is due to the peculiarity of the course that our argumentation follows. It is undoubtedly the result of the study of many texts and ideas by illustrious thinkers, but even their ideas get so twisted and stretched – though within the limit of an honest interpretation – and then recombined together that it becomes difficult to recognize someone in particular to whom assign the reference. As regards the (huge) lacks – e.g. leading philosophers omitted, or important parts of their thought not considered – that one could find in that chapter, our defence resides, once again, in the peculiarities of the course that we follow. It does not claim to be the handbook of ontology on both sides of the boundary between philosophy and computer science. Rather, it is an original reading of ontology that bridges that boundary, with the intent to enrich the understanding of the current issues of ontology in both fields by means of insights and suggestions borrowed from each other.

After such a "disclaimer", we may then acknowledge the inspiration of the ideas expressed in the next chapter mainly, but not only, from the works of Smith ([Smith, 1999, 1995, 2006, Smith and Welty, 2001, Smith, 2003]), and Nef ([Nef, 2009, 1991]), while retaining to our responsibility the identification of the moments, and of the philosophers, who have marked the shifts and turns that we will signal in the history of ontology.

Though uncanonincal, we believe that such a work is yet fully in the line of philosophical research. In fact, it cannot be counted among the works that proceed in verifying the trustworthiness of an interpretation of the thought and the system of a given philosopher; nor it can be counted among those that unveil an original interpretation, well founded and maybe motivated in the discovery of some unpublished manuscript; nor even among those that collect and compare all the positions about a delicate issue, maybe highlighting similarities beyond seeming divergences. Rather, our work humbly aims to be counted among the works that stimulate discussion and claim for further investigation by raising attention on original issues, or on old issues but from an original perspective. Inevitably such an effort brings also a first interpretation of the subject that, due to its freshness, may be not exhaustive, incomplete and perhaps could even be found to be flawed somewhere. Though, we proceed in this work with the aim, and the awareness of, not to say the last word on the subject, but rather the first one, at least for the particular setting of the question that we propose. Indeed we obviously could not say the first word on ontology *tout court*, but we believe that we are tackling for first a thorough study of the intricate relationship between reality and its (symbolic) representations as seen from the fascinating viewpoint of (Semantic) Web. From there, one can get precious insights about communication, knowledge, ontology and perhaps even reality itself. In the following we will do our best to give the reader all the hints and clues that make us reach for our conclusions and convictions.

7. Current readings of the history of Ontology

Nowadays, when searching in (almost) whatever dictionary for the word *ontology*, one finds two main acceptations considered: the one strictly philosophical and the other noted sometimes as logical or directly as relevant to computer or information science. The corresponding definitions range from the concise pair of an on line dictionary[1]:

1. "*Philosophy* - the branch of metaphysics that deals with the nature of being"

2. "*Logic* - the set of entities presupposed by a theory"

to more articulated definitions like those of a renowned dictionary[2]:

1. "A science or study of being: specifically, a branch of metaphysics relating to the nature and relations of being; a particular system according to which problems of the nature of being are investigated; first philosophy"

2. "A theory concerning the kinds of entities and specifically the kinds of abstract entities that are to be admitted to a language system"

In these latter and more complete definitions we get some more elements. In particular, the first defintion contemplates multiple acceptations slightly different, yet in the philosophical domain, which include, besides that branch of philosophy first "isolated" by Aristotle, also the ontology intended as a philosophically grounded position based on which one may investigate the (other) aspects of being as they are approached on particular respects in the other sciences – that is, ontology as the collection of ontological assumptions on which the sciences, or even the rest of the system of a philosopher, can be based. On the other hand, the second definition perfectly fits the idea exposed in the first part of this work about the ontological commitment of an artificial language designed to represent the knowledge contained within an information system. Lying behind these definitions, however, there are the two ideas that have already motivated the distinction, in the first part, of an Ontology (in singular form and with initial capitalized) from many ontologies (in plural form and with lowercase initial).

In spite of this usual neat distinction, however, we could also remark the commonalities that make both to be named by the same term. Indeed, even though the secondary acceptations are referred to the specialized domains of logics, computer science or information science, they say something that holds also for natural languages. After all,

[1] The Free Dictionary by Farlex, `http://www.thefreedictionary.com/ontology`
[2] Webster's Third New International Dictionary

Quine has not proposed the criterion for ontological commitment thinking of artificial languages for machines, albeit he did it for languages with some logical regimentation (we will come back on this later in this chapter). Apparently the point is in the degree of precision of the language, typically understood and reduced to, or produced by, the formality of a logical language – just think of the many efforts in the quite recent history of philosophy for a "pure" language able to emend argumentation and debate of any misleading interference depending on the complained inaccuracy of common, natural language to deal with the deep reality of things, especially in the field of ontology! As a consequence, an ontology (as a system) presented in a purposely set up flawless language could be a *formalized ontology*, not only those for use in computer based information systems. On the other hand, the possibility to consider an ontology as the (more or less) definite, definitive and exhaustive classification of beings (in any form of being possible in a given world) holds both for philosophical ontological inquiry and for Knowledge Representation. So that the choice to name ontologies that particular form of knowledge representations which is used also in Semantic Web (but not only there), which at first may sound like pretentious, is after all adequate. The differences between Ontology and ontologies, then, can be counted in

- the degree of formalization of the ontology, ranging from original speculation in natural language to the artificiality of a language standardized by an international organization and implementing a Description Logic (think of OWL for instance)
- the scope of the ontological inquiry, which is very limited for Semantic Web ontologies, whose metaphysical realm, we would almost dare to suggest, is just our common world
- and clearly the purpose of the ontology, which is on the one hand the ambition deeply rooted in human nature of discoverying the ultimate answers to the great question of life, the universe, and everything; and on the other hand just have computers properly handling information.

Now, before we get in a more detailed comparison and, most of all, before we suggest which contributions and insights our experience with Semantic Web ontologies – and the other forms of knowledge organization for the Web – could bring to the philosophical approach to ontology, we will propose a quite singular rush course into the history of Ontology by touching just a few of prominent philosophers and signalling the turns in the approach to ontology that we deem as the most significant, at least with respect to the aspects that we are going to match with our "findings".

In fact, although the effort to precisely define what an ontology for the Semantic Web is has taken much time and many pages in the first part of this work, it is nothing compared to what would be needed in order to give an adequate account of the philosophical history of ontology. As a consequence, given also that this work meets philosophical ontology (only) for a (mutually we believe) profitable comparison and is not focused on the history of the thought about ontology up to most recent developments in "applied ontology", we cannot but walk an original path that will be surely not exhaustive nor complete, but at least illuminating and philosophically honest.

We will start with Aristotle and his "first philosophy", or science of being *qua* being, because it is he who has "invented" ontology as a special branch of philosophy, with its specfic objectives and method. By the way, it is interesting to note that Aristotle never used the term ontology to name it, albeit it is a word originating from Greek. Such a fundamental part of philosophy, indeed, has stayed without a definite name until the beginning of XVII century: the coinage of the term *ontologia*, latin for ontology, expressing in one word the concept of "study of the being" happened around the year 1613 at the hands of the German philosopher Jacob Lorhard (known also as Lorhardus). The term then got spread and largely known thanks to the succeeding works of Christian Wolff, another German philosopher who dedicated himself mainly to ontology. Part of Wolff's ideas are later recognizable in Kant's thought, albeit Kant is considered one of the most hard critics (even opponents) of the metaphysical thought, of which ontology is a fundamental part. But, in order to avoid continuous "flashbacks" in the history of philosophy, let us start from the beginning and track by main steps the development of ontology over the last twenty-four centuries.

7.1. Aristotle

We start with Aristotle (IV century B.C.) since he is the first to recognize the need for a special science devoted to the identification of the most basic, fundamental and general characteristics of every form of being, precisely that which he calls the "science of the being *qua* being". It is indeed the study of the being as far as just it is, with no attention to any other possible respect under which one could consider the being – like for instance as a living entity, or as an object moving in the space. Such additional respects are the focus of interest for other sciences, biology and physics for instance. And by the way, also biology is one of the sciences to which Aristotle has largely contributed by establishing much of the method and terminology which have been used for centuries after him.

Nevertheless, Aristotle was not the first to feel the need to study the being as such. About a century before him, Parmenides – among the eldest philosophers whom the history of occidental philosophy has information and testimony about – had already posed the question of how can one speak of the being in its totality unless getting in some specific aspect of a particular being and, as a consequence, falling in the absurd of conceiving not-being. After Parmenides and his quite cryptic idea of the being – which finally was more about one single perfect, immutable Being (capitalized) totally distinct from any form of being that one could ever meet in human life – the history counts also other Greek philosophers who dealt with the issue of what is the being, for instance the so-called *atomists*, whose ontology (*ante litteram*) was a materialistic one based on the idea of atoms and vacuum. After them, and again with an ontology quite detached from the material world (like in Parmenides), there is Plato. He indeed conceives the real being as something that lies in another world, that of the Forms of which, in our present world, we can just perceive distorted images.

The three positions quickly mentioned here above, that is the conceptions of the being expressed by Parmenides, the atomists Leucippus and Democritus, and Plato were not strictly speaking ontologies – or at least not with respect to nowadays acceptation of the term ontology as used in philosophy. They were rather, in nowadays terms, metaphysical positions. But here again, speaking of metaphysics, we must call Aristotle into play in order to properly define metaphysics along with ontology. Thus, finally, it is clear why his figure appears so fundamental in the history of philosophy and worth being taken as the starting point for our rush course. While the term *ontologia* has been coined with no direct reference to Aristotle – albeit we have already said that it stands precisely for that study of the being whose methods and basic categories have been first systematized by Aristotle – the term *metaphysics* has been coined, in a time much closer to Aristotle's activity, by ancient scholars who needed a new term to call that part of Aristotle's writings that had no specific title and which stayed "unnamed" also after that ancient "librarians" had edited the aristotelian *corpus*. Scholars then started calling it *meta-physics*, i.e. after the (writings on) physics simply because those writings had been placed after the ones about physics, physics standing generally for "about natural world".

Nevertheless, based on the kind of arguments presented in those writings, the term metaphysics has entered the philosophical vocabulary as the name for that branch of philosophical activity that raises the speculations of human thought up to the maximum heights, facing questions ranging from the origin of the universe to the existence of God, to the actual reality and nature of the world, to the attributes of being, and so on – thus configuring also a reason to understand *meta-physics* as meaning beyond physics, beyond the natural world of which one can have a more or less direct experience.

Ontology then can be seen as comprised within metaphysics, as it actually occurs in Aristotle's writings called metaphysics. Ontology indeed provides a neat and clean preparation of the field of metaphysical work by allowing for the identification of all that exists, so that further metaphysical speculation may focus on the nature, the causes, the purpose (or end, goal) of any given object whose existence is granted by the ontology. It must be noted, however, that this picture is reasonable and works quite fine as a provisional explanation sufficient to make us pass over and touch next points of the discourse; but it is pretty simplistic because it hides the fact that also ontology requires that at least part of the other metaphysical issues have been previously set out, so as to have reliable criteria to identify that which (really) exists. This will be an interesting point to be discussed with respect to ontologies used as knowledge representations in information systems. But for the time being we may content ourselves with this simplistic position. After all, the purpose of ontology as the science of being *qua* being in Aristotle is to reach for the pure, absolute being and exhibit its characteristics before passing to the study of the single beings by means of special sciences. This is the reason for the appellation of "first philosophy" that Aristotle pays to the science of being *qua* being. We need just to bring with us the consciousness that any ontology is marked by some, more or less intuitive, metaphysical assumptions.

Let us briefly introduce now the fundamental points of Aristotle's ontology and show the originality of his setting of the matter, along with the main topics that have become inescapable issues for subsequent philosophers. It is noteworthy, indeed, that still nowadays Aristotle appears as a valuable term of comparison for the philosophers involved with ontology, who are for the most part in the stream of analytic philosophy.

7.1.1. The Categories

Since the purpose of his first philosophy is to observe beings only for their being, Aristotle provides criteria and categories (we can say just categories for sake of brevity) that allow to recognize the features proper of any being – note that he introduces these criteria and categories in another writing than the so-called Metaphysics, named (unsurprisingly) *Categories*. The metaphysical assumptions that drive the subsequent ontological work, and which stay implicit in Aristotle argumentation, are embedded precisely in such categories. Briefly: Aristotle adopts one particular point of view on reality and then also his categories, supposed to isolate the tracts constituent of any being, reflect that point of view. The fact, then, that Aristotle's categories have stayed almost unchanged for many centuries and assumed as good "lenses" to observe reality, we argue, is due partly to the perspicacy of the categories that Aristotle has selected[3] and, perhaps most of all, to the fact that they have been the first to be proposed as a method to investigate reality. Indeed, it is easy to agree with aristotelian categories as far as they may appear to be compliant with the intuitive understanding and knowledge of the world, which is a point largely missed by the metaphysical settings of his predecessors – although these last were not necessarily wrong, as for instance the atomistic theory of Leucippus and Democritus has been largely verified (much later) as soon as phyical science has developped enough to be able to "see" atoms.

It is time to put some irons in the fire, otherwise it would be hard to discuss deeper issues concerning the aristotelian ontology, like for instance the relationship holding between the categories and the beings, single entities, without having at least sketched it. Based on remained writings[4] we have at least two main schemes for the classification of beings into categories which have been proposed by Aristotle – plus a series of alternative metaphysical foundations for his system that testify both for the fecondity of Aristotle's thought and for its progress over a long time in which re-thinking of many aspects has surely occurred, with the difficulty for us to determine with certainty which writings states the last version of Aristotle's thought.

So, we have to account for both the main categorial systems. The first that one meets in the *Categories* allows for a classification of beings based on two properties such as *to be said of something* and *to be present in something*. These two are orthogonal and each of them may be or may not be true of any given being, so that the resulting scheme is articulated in four categories

[3]To be honest, the writings by Aristotle that have remained to us leave some confunsion concerning the foundation, and then the soundness, of his categories. It is thanks to the interpretative effort of many ancient and medieval scholars (among which the most influent has been Thomas Aquinas) that Aristotle's categories have been formed to really compose a system. Such a system, however, might even be somewhat different from the original ideas of Aristotle, medieval interpretations having been sometimes pretty "free".

[4]And omitting here any discussion about the authenticity of those writings.

1. not-said-of & not-present-in
2. not-said-of & present-in
3. said-of & not-present-in
4. said-of & present-in

Aristotle gives only a quick explanation about these categories and some examples, and the question of the right interpretation of this scheme remains an open issue. Nevertheless, given the nature of our crash course we have to jump directly to its most largely accepted interpretation. It is the one developed in the middle age which associates to the four categories generated by the combination of the two properties the following classes of beings

1. non-accidental particulars (individual substances or prime substances)
2. accidental particulars (somewhat similar to *tropes* in today's terminology)
3. non-accidental universals (universals *tout court* or second substances)
4. accidental universals

The fundamental category is that of non-accidental particulars, i.e. the "pure" particulars, which are the individual beings like Socrates, the reader of this work or the particular chair on which the reader is sitting. These are the *substances*. Substances, which are existing individuals (or concrete objects), are the elementary objects of Aristotle's ontology – which appears to be markedly concrete, focused on sensible objects, as opposed to Plato's ideas, which were mainly focused on ideal forms out of the reach of human senses. Aristotle's substances (to be precise: *prime substances*) cannot be said of any other substance nor can be present in any other substance, as well as neither Socrates is present in any chair, nor the other way round, nor Socrates can be said of the reader – and by the way what would it mean to say Socrates of someone else? Rather, any other form of being, accounted for the other three categories, can be said of and/or be present in some substance. So, for instance, *man* can be said of Socrates, but it is not present in Socrates. It is a non-accidental universal, or a *second substance*; briefly, it is a kind of being for the particular being named Socrates. Moreover, to be a man is essential to the nature of Socrates, in this sense it is non-accidental. Yet, there may be some *knowledge* in Socrates, e.g. the knowledge-of-not-knowing: such an individual knowledge is particular to Socrates, it is, according to Aristotle, present-in Socrates (it is an accident); but knowledge-of-not-knowing cannot be said-of anything, not even of Socrates himself – i.e. it is not a kind for substances – so it is called an accidental particular. Finally that which can be both said-of and present-in some substance is called accidental universal, like *knowledge*, which can be said-of Socrates' knowledge-of-not-knowing – after all it *is knowledge*! – and is also present-in some other substance, i.e. for instance into Socrates.

Although the argumentation may look like a little muddled as it is sketched here, the categories of universal, particular and sometimes also that of accidental particular (now also known as *trope*) are used still nowadays in many (most) works on ontology.

Besides this four-fold classification of beings, Aristotle provides also a second set of ten categories. The first category is, again, that of substance, explicitly named as such in this scheme. It is apparently the cornerstone of aristotelian ontology. The other nine categories are *quantity*, *quality*, *relatives*, *somewhere*, *sometime*, *position*, *having*, *acting* and *being acted upon*. We are not going to examine all of them, nor a few, since it would require to enter major issues about Aristotle thought and writings which have been the "battlefield" of scholars and interpreters for centuries up to now. We will rather sum up the few aspects on which most of them agree. In particular, once isolated the category of substance, all the other categories can be seen as ways of being of substances. Moreover, like the category of substance collects, as the highest kind of substance, any other kind of substance[5], every other category is intended to be the same for the many kinds of things which are not substances. For instance, the category of quantity is the highest kind of measure for substances, whereas the category of quality is the highest kind of attributes for substances, and so on.

This second scheme of classification then seems to propose an alternative to the first one. The set of the ten largest, maximum kinds of things (substances and their modes) indeed provides a "grid" where to place substances and, in principle, anything that affects substances, discarding the distinction between things that can be said-of substances and things that can be present-in substances, but also implicitly taking into consideration only the said-of property – possibly by stretching the argumentation in the *Categories* in such a way to have things, which would have been counted as present-in according to the first scheme, fit into the alternative classification that cares only for things that can be said-of some substance. We must note, after all, that the word "category" derives from the Greek word for predicate, that which clearly suggests a guiding interest for what can be said about beings. This is the first hint concerning the relationship between ontology and language that is a sort of *leitmotiv* of this work. We will insist more on this in the following; for the time being we just note that with Aristotle – usually recognised as the "father" of ontology (for all that we have already considered) – it emerges this relationship, which does not appear in his predecessors' argumentations about the beings, the Being and its very nature.

Together with the list of the largest kinds for classifying beings (and other things, like those which can be said of beings), the aristotelian system gives us also the method for operating an internal classification of kinds. If one considers more broadily the philosophical production of Aristotle, from the works on nature (physics and biology) to those on metaphysics and logic, it is quite easy to derive the internal classification for some of these categories. For instance, this is part of an internal classification of the highest kind of substance (second substance) that Aristotle surely would agree on (cf. [Studtmann, 2008]):

Substance

1. Immobile substance
2. Mobile substance

[5]Note that the category of substance here corresponds to the second substances of the first scheme, i.e. to the universals (that which is said-of but not present-in), the kinds for prime substances.

- Eternal mobile substance
- Destructible mobile substance
  - Unensouled destructible mobile substance (Elements)
  - Ensouled destructible mobile substance (Living things)
    - Incapable of perception (Plants)
    - Capable of perception (Animals)
      - Irrational (Non-Humans Animals)
      - Rational (Humans)

Such a classification actually contains a hierarchical taxonomy like that we have talked about in the first part of the present work – or, rather, it may better be the case that the work of Aristotle has introduced such a way of organization of knowledge, and of reality of course! Since indeed Aristotle was committed to present the actual organization of the world, not (only) human knowledge of it. It is the same technique that Aristotle adopts for classifying living species too and animals in particular – in a classification that, by the way, has proven really solid resisting until the XVIII century and the binomial system proposed by Carl von Linné.

Two features of the Aristotle's system of classification, which underpins his ontology, are to be noted. The first one concerns the way how the classification scheme is matched against reality and requires the classified objects to be assigned to a single kind (or species). For instance, in the branch of classification illustrated here above, a given substance like the individual Socrates should be assigned to the farthest possible node from the root that suits for it, which for Socrates is with no doubt "rational ensouled destructible mobile substance", or shortly rational animal. Clearly, as we are accustomed with ordinary taxonomies, also in the aristotelian ontology any instance of the kind rational animal inherits the upper kinds, that is the kinds along the path that connects the root to the considered node. Then, Socrates is also an "ensouled destructible mobile substance capable of perception", a "destructible mobile substance" and finally a "substance". In this sense the ten *genera maxima* – the highest kinds – of the second classification scheme by Aristotle are the highest: they collect, by inheritance, all the individuals that belong to terminal nodes of the classification tree. Moreover, there is no single highest category, no *summum genus* that contains the ten categories, so that the *maxima genera* have no common root. It is precisely the disjunction of these ten branches that allow for a given prime substance to find its place within the category of (second) substance and also within any of the other nine. That which results in the possibility to predicate, according to Aristotle, a place, a time, a position, a relative, a quantity, a quality and so on for any given particular being, i.e. prime substance. Note however that, apart from the category of substance, it is not needed to classify every single (prime) substance within all the categories. With this in mind, we must note that the the aristotelian classification system offers much more than a hierarchical taxonomy. The ten categories indeed offer ten parallel taxonomies for classifying beings according to multiple points of view: based on their essence – the second substance – which produces the classification with major ontological significance, but also based on their position, habits, relations with other substances, etc.). Therefore, already considered the inventor of so many things, Aristotle, we think, should be credited also of the invention of the facets system for catalogation.

It is now time to fine-tune our terminology: we must signal that only terminal nodes in the aristotelian classification system are strictly speaking types, i.e. *species*. Any kind which can be further analyzed and specialized in some other types is a *genus*. After such a terminological precision, we may consider the other most relevant feature of the Aristotelian classification system. It concerns the way how species are characterized, or defined, which happens according to the technique called *per genus et differentiam*, which indeed is a technique first attested in Aristotle's writings. Each genus (kind) may be specialized in possibly many sub-genera that, in their turn, can be furtherly specialized in other genera or stay as terminal nodes of branches – thus playing the role of species. The specialization of genera into species (or sub-genera) is produced by selecting the relevant genus of which the species is a specialization and signalling the characteristic for which that species is different (the *differentia*) with respect to the whole genus and to the other sibling species (if any). No need to remark that the same mechanism is into play also in typical hierarchical taxonomies. Consider for instance the example above: the genus Animal is specialized in two species, rational and irrational animals respectively, in such a way that a Human, i.e. a rational animal, is an Animal (superior genus) whose species is distinct from that of the other animals based on the characteristic property (the *differentia*) of rationality.

Yet another relevant point about the aristotelian system is the importance attached to the language, which appears again a novelty with respect to previous philosophers. Indeed, just before introducing the two systems, Aristotle discusses in *Categories* what kind of assertions are to be credited with the ability to predicate and therefore to be true or false. After all, we have already remarked that the term category derives from the Greek word for "to predicate", so that it seems not surprising that we find in Aristotle and his science of the being *qua* being also the origin of the tight relationship between ontology and language. It is precisely the derivation of the categories of his ontology – as an organization of beings – from the ways how things can be said, predicated of beings that which produces the connection between language and ontology in Aristotle, and since Aristotle until now.

In particular, the observations that Aristotle proposes about the language are concerned with the identification of what kind of expressions are actually predications, and are strictly preparatory to the introduction of the two schemata for classifying beings. As a consequence, considering the largely accepted interpretation of the first schema consisting of the four categories of (in nowadays terms) universal (non-accidental or accidental), particular, and trope, we have that universals, which are grounded on the property of being said-of some substance, are all the expressions that can be true or false of some being.

Now, after having somehow celebrated Aristotle for so many and crucial contributions to the western approach to ontology, we should also put all this back in the right perspective in order to preserve intellectual fairness and to respect historical facts. Then, since this work does not provide us with room enough to correctly settle the historical figure of Aristotle nor to fully appreciate the role of subsequent interpretations on his thought – the medieval ones having been even very "enriching" with respect to the base of remained writings – we must at least point out some criticalities in the picture that we have just outlined and try to indicate the minimal coordinates of the historical placement of Aristotle's ideas.

First of all we have to remark that we have not herited a single, complete system about ontology from Aristotle, since remained writings – beyond questions about their authenticity – contain many different, alternative accounts for same issues, which most likely are the result of changes in Aristotle's thought during the long period of his scientific and philosophical activity. An additional factor of complexity in such a situtation is the uneliminable uncertainty about the right ordering of the writings, which is most apparent just with the books that have been collected under the title of *Metaphysics*. So that it is not easy – and really it is quite controversial – to identify what is to be taken as the final position of Aristotle about ontology and the ultimate nature of beings. For instance, some difficulties can be found even within the *Categories*, for the two schemata look like complementary to each other, but no clear relation between them is established by Aristotle. They seem just juxtaposed one after the other and, even though the non-accidental particulars (prime substances) fit well in the highest kind of substance, none of the other kinds deriving from the four-fold schema finds a comfortable place in any of the other nine highest kinds. Nor Aristotle suggests how to consider one schema with respect to the other.

Nevertheless, we can observe, by contrast, that the category of substance is the only which is paid a great attention by Aristotle: it emerges with the evidence of a matter of fact that it is the core of Aristotle's ontology and more generally of his metaphysical system. Indeed, the category of substance not only can be clearly recognized in both the schemata (the four-fold and the ten-fold), but also on the notion of (prime) substance – an inseparable combination of matter and form – are founded other key elements of Aristotle's metaphysics, from the theory of causes to the nature of the First Cause, and the necessity of its existence.

The deep analysis of the category of substance and its crucial role in explaining the fundamental mechanisms of the nature – up to the definition of a cosmological theory where the First Cause acts as the Unmoved Mover of the whole celestial "machine", and indirectly as the final cause of every form of *change* – allow to identify the role of ontology as the *first philosophy*, whose task is to determine what exists in the world, in the universe that one wants to observe and study in its deepest, as the primary element within the field of metaphysical inquiry, which not by chance gets its name after Aristotle's writings, so that the other branches of metaphysics will deal with the clarification of what ultimately is, in its nature, that which exists. But, on the other hand, such a focus on the notion of substance also makes to regret even more the absence of a similarly rich argumentation concerning the expediency and plausibility of the other nine categories that Aristotle proposes.

However, just after Aristotle many commentators and interpreters of his works have arranged the missing explanations. Perhaps the richest and most clarifying explanation is the one provided by Thomas Aquinas, who indeed is among the most influential authors responsible for the composition of the trails of Aristotle's thought into an organic metaphysical system. In fact, without going too deeply in the subject, we just note that remained writings from Aristotle miss an overall, exhaustive explanation of the metaphysical assumptions that he takes in order to identify the ten categories as highest kinds, and also when he organizes their inner taxonomy.

Concerning the ten categories, indeed, Aristotle introduces them but does not explain the reasons for which they should be the "right ones", nor he explains from where the categories are derived. To be provocative one might argue that, Aristotle being the first to propose a system of categories, he could have thought of it as the only natural and reasonable way to account for the beings in the world and left to his opponents the task to prove the contrary. With this provocative idea in mind it looks quite surprising the fact that aristotelian categories have stayed for such a long time – up to XIV century for sure – an inescapable and irreplaceable term of comparison for any theory or position concerning the categories to organize reality. Without being provocative, those categories may appear as at least questionable, and indeed it has been one of the main issues about the theory of the categories throughout the middle ages to determine the right number of highest kinds, or ultimate categories, the fundamental point being the difficulty to determine whether other more general categories, therefore including at least some of the aristotelian ten, could be conceived and/or individuated. Or, on the contrary, more *maxima genera* were to be considered. Indeed, even if his ten categories might have appeared so natural to Aristotle, it has not been necessarily true for his commentators from the ancient times to renaissance throughout the middle ages. The heated debate which has developed over centuries about categories and their number has seen many different positions, envisaging also different sets of categories based on a deep criticism against the possibility of deriving precisely the Aristotle's ten categories as highest kinds.

Concerning the taxonomy within any single category – that is the specialization of genera – another kind of questions raise concerning the entities which are assumed by Aristotle in his ontology – precisely because they are part of the internal articulation of some category – but which are not clearly introduced, justified on the metaphysical plan so as to convince the reader of their existence, or at least of the expediency of assuming their existence. Consider for instance the case of numbers, introduced by Aristotle as a kind of discrete quantity in the category of Quantity but whose ontological status, their very nature stays mysterious. The only certainty is that they are not substances. As we have mentioned above, however, medieval commentators and interpreters have largely arranged – sometimes by means of quite "personal" interpretations – the aristotelian categories providing them with a background sufficient to justify their derivation and the correctness of the number of ten for the highest kinds, albeit others have refused this number either proposing less categories or more, within a long lasting, sometimes heated, debate. Perhaps the best accomodation in the line of the defence of Aristotle's original setting is that by Thomas Aquinas, which by the way is the one that we have adopted, on the sly, when describing the categories other than Substance as categories of ways (modes) of being.

We think, however, that much of the debate on Aristotle's categories would be better approached by properly weighing the only argumentation that Aristotle gives to support his theory of categories, that is the considerations about predication and the language. Based on this, it is quite clear that in fact categories are just forms of predication about beings, so that the interest in the nature of the kinds of the nine categories apart from that of Substance, although fair on the part of commentators, is not an interest proper to Aristotle. To him they do not seem to form categories of *per se* interesting beings, since the only interesting beings are the ones accounted for in the category of substance. Indeed, Aristotle's ontology appears focused just on concrete objects, the ones which are substances, whereas the modes of predication recorded in *Categories* are not considered as other beings, to be classified along with substances, but rather as just things that can be said-of substances. Yet, some objections could be raised when considering particular accidents (nowadays tropes) which may seem somewhat border-line entities, but it is clearly out of Aristotle's interest in *Categories*. It is rather an interesting point for subsequent and even more a crucial point for the current research on ontology.

Because of this shift on predication we are back to the crucial point: the connection between ontology and language, which actually shadows a larger – and even more problematic – relationship involving language, mind and reality. The *Categories* indeed has played the role, through the plurality of commentaries and interpretations of which has been the object and together with many other of Aristotle's writings, of opening and initiating a variety of issues which have proven extremely fascinating for subsequent philosophers up to now, and most of the critical issues that concern ontology and metaphysics are after all connected to that triple relationship involving language, mind and reality which firstly has emerged from the work of Aristotle. By the way, the triangle whose vertices are *concept*, *symbol* and *object*, well known in the field of Knowledge Representation and improperly called "aristotelian triangle" – since it has been really drawn only relatively recently (at the beginning of XX century, cf. [Ogden and Richards, 1923]) and then, later, backward christened thus – actually crystallizes that which is the mainstream interpretation of Aristotle's position about such a relationship, based mostly on the interpretation by Porphyry. In fact, quite openly within the concise argumentation about the language in the *Categories* – but also in some other of his writings (especially in *Metaphysics*) – Aristotle expresses the conviction that the words of the language stand for objects in reality, what is especially apparent for proper names of (prime) substances, but he also holds that even predicates *are* things, that is object with some form of existence. It is apparent, for instance, when he introduces what we now call universals, and he writes

> Of things themselves some are predicable of a subject, and are never present in a subject [Aristotle, 2000].

so that what is predicable is a thing itself, liable to be classified, as indeed Aristotle does within the first schema, the four-fold one, by defining the class of things which are predicable of a subject (i.e. such that can be said-of) but are not present-in. Nevertheless, the second schema, for which the name of categories stands properly, does not account for such a kind of things. Rather it accounts only for substances as things-in-themselves, whereas all other categories are considered as far as they are ways of being of substances, that is in particular predicates about substances, so that the interpretation by Porphyry has strong reasons to focus on the language and the predication as something distinct from actual reality of beings. Moreover, while the direct relation between language and reality – perfectly "touched" in the case of (proper) names of (prime) substances – is openly tackled by Aristotle, the third vertex of the so-called "aristotelian triangle", i.e. the concept in the mind, is not explicitly dealt with by Aristotle, not only in the *Categories* but also in general in all of his writings. We can just signal that some hints on this issue (especially in *De Interpretatione* [Aristotle, 1928]) show Aristotle's awareness that a spoken sound, an utterance, symbolizes some mental experience which is common, better: the same, for everybody (who speaks a given language) and such mental experience in its turn is the image of things which are, once again, the same for everybody. Although it is not called *concept*, the mental experience that Aristotle talks about seems to play exactly the role that concepts in the mind play in the contemporary "aristotelian triangle". Nevertheless, Aristotle does not say much more than that: he does not provide any explanation of how words, and in particular predicates, relate to reality, nor about the nature of the things that words symbolize, so that it can be easy to grasp how it should work in the case of the nouns which are names of substances, whereas it stays somewhat mysterious what should be the things – which are supposed to be the same for everybody – which the predications concerning the ways of being refer to.

The lacks in Aristotle account of the language and its relationship with ontology, like e.g. a thorough metaphysical account comprising also "mental entities", should not surprise. Indeed, even though Aristotle first pays attention to the language – mainly as the instrument to classify beings according to their ways of being identified by means of the ten categories which cover all the forms of predication about substances (the basic elements of Aristotle's ontology) – his effort is not to be confused with or mis-compared to the strong interest on language with respect to ontology which has become a key element in contemporary analytic philosophy and which is partly derived from the efforts of logical positivists to attain a pure language capable to eliminate any form of ambiguity from the philosophical debate – albeit they would have refused to speak of ontology inasmuch as it is usually considered together with and within a metaphysical theory about the world and the universe.

The interest of Aristotle about the language, we think, is not absolute, it is rather directed to provide an original setting for the central problem in the philosophical debate of ancient Greek philosophers, that is the problem of the being and the change. The Being, as perfect and immutable, on the one hand, and the continuous, unending change on the other hand had been posed as the basis of their philosophical systems by respectively Parmemides and Heraclitus more than a century before Aristotle. Plato, among others such as the atomists Leucippus and Democritus, had proposed his way to reconcile the two principles. To put in a nutshell he had postulated the existence of the *forms*, perfect and immutable version of any being of which one could ever have experience in the world. However, the beings encountered in a human life, according to Plato, are not the forms, which belongs to another world (the *Hyperuranium*), but only a more or less close copy (a shadow) which in any case is a corrupted version and, as such, it is liable to change.

Aristotle, who had been a student of Plato in the Academy in Athens, later entirely refused Plato's system. That which we have tried to sketch here above, therefore, is the part relevant to ontology of the huge work by Aristotle aimed to set an alternative system that does not put reality – the true beings – out of the sensible world. In Aristotle's system indeed there is no alternative, superior world; nor are there different degrees of being (as it was in Plato's, with perfect forms and imperfect shadows); there are only the (prime) substances, whose essence (i.e. the natural form immanent in them) is perfect and immutable although their matter is liable to change due to the process by which any substance naturally develops and finally attains the mature stage of its natural form. The interest of Aristotle about the language then should be understood in such a context, where it is the instrument that allows to deal with the aspects, processes and phenomena of change which are observable over and in substances, but which are not substances themselves. Only substances, roughly speaking concrete objects, appear then as stricly speaking *beings*, whereas all the rest is *ways of being*, accountable for on the part of language.

Now, having largely – with respect to a rush course – talked of Aristotle, for he provides the paradigmatic case for our discussion of ontology and its relationship with language, we will now on proceed more quickly with the rest of our selction of philosophers most significant for the evolution of the thought about ontology.

7.2. From Porphyry to the Scholastics: the problem of universals

The canon of western history of philosophy presents us with a list of relatively important names and schools after Aristotle. These however focused their interest mostly on ethical and moral issues rather than on general metaphysical and ontological questions, so that we jump directly to Porphyry (III century C.E.). Actually Porphyry does not propose a whole original system nor an alternative approach to that which seems to be the central problem for ancient Greek philosophers – that of the being and the change. Rather he is concerned with the harmonization of the system of Aristotle with the thought of Plato, after his master Plotinus had opened the way for a reinterpretation of Plato – the so-called *Neoplatonism*. After all, the standard way of studying "the classics" then followed in the Platonic Academy was precisely to use Aristotle to reach a full understanding of Plato's thought, even through Aristotle's criticisms to Plato. Then, surely, Plotinus would deserve more attention than that we are going to tribute him here, since he also proposes something quite original with respect to Plato's metaphysical setting, albeit he intended to produce just the right interpretation of Plato. Nevertheless we are more interested in his student Porphyry for the extent to which this latter brings the reinterprtation of Plato, that eventually produces very interesting results regarding ontology.

As we have already had the occasion to note, in the late antiquity and during the middle ages the standard way to practise philosophy was to interpret and comment on predecessors' writings, most of all those by the greatest figures such as Plato and Aristotle. In this trend is to be considered the role of Porphyry, who nevertheless offers to the development of philosophical debate some truly critical elements. The originality of Porphyry's production is apparent also in his effort to read in the new light of the revised Platonism (or Neoplatonism) many of then current issues, like for instance in the religious field with an original conception of the role of religion and religious rites.

Anyway, we are here interested in particular with one text by Porphyry, the *Isagoge*, that is the introduction to Aristotle's *Categories*, or more generally to his whole ontological and logical thought, with Porphyry's rich commentary. It is important to us for it introduces that which will become a central theme for debate up to the modern age, and again from the XX century specially among contemporary analytic philosophers, that is the *problem of universals*. In *Isagoge* Porphyry tackles the point, that we have already remarked while talking of Aristotle's *Categories*, concerning the ultimate nature of the categories and of the "entities" that should be supposed lying behind them. Indeed, even though we have argued that such a kind of problems were not of primary interest to Aristotle, who most likely was just interested in properly setting the problem of being and change, the ontological status of the categories and other elements that he had posited to build his system requires to be clarified and clearly defined, especially when – as it was the case – there is a plethora of commentators and interpreters whose activity is (supposed) to make clear any possible obscurity.

Given the introductory strain of the work, in the *Isagoge* Porphyry just states the problem. Most likely he proposed also some kind of solution to the problem in some other lost writing (a second longer commentary on *Categories* is known to have gone lost except for a few fragments). In any case, the issue that Porphyry poses concerns three fundamental questions:

1. whether second substances exist;

2. if they exist, whether they have a bodily form of existence;

3. if they have not, whether they are separable from sensible objects.

Once clearly exposed by Porphyry, the problem has been given the dressing with which it is widely known (think for instance to the Latin word *universalia*) thanks to the translation and commentary of Porphyry's work by the Roman philosopher Boethius (V-VI century C.E.). Indeed, Boethius's translations of Porphyry and Aristotle have been fundamental textbooks for the education in logic of generations of philosophers, and more generally of learned men, throughout the middle ages in the education system promoted by Charlemagne, that will result in the Scholastic[6], from which first universities have been born in Europe.

From Boethius we recover both a richer setting of the problem of the universals and also his personal attempt to solve it. At the time of Boethius indeed over other two centuries of interpretations and commentaries had sedimented upon those questions and one standard way of approaching it had been established, according to which universals can be considered

ante rem (before the thing) that is universals exist and have their own form of existence separated from bodies;

[6]Considered as a system of teaching coupled with its specific method consisting in dialectical confrontation based on a given text by some good authority (the *auctor*).

in re (in the thing) that is universals exist insofar as they are in some body;

post rem (after the thing) that is universals emerge as an abstraction from bodies.

It is quite straightforward to recognize in the *ante rem* position an understanding of the issue compatible with the Platonic conception of the being. Indeed, according to Plato, pure *forms* do exist *per se* in a separate world, and there are forms for anything we can understand – since actually, according to Plato, we can recognize and understand a thing (the *res*) as far as we are able to recognize in it the shadow of some form that we have already encountered before (*ante*), maybe in a previous passage of the soul through the *hyperuranium* world of forms.

On the contrary, the *in re* position can, roughly, be compared to Aristotle's conception of ontology since universals, his second substances – and the other properties that can be predicated about prime substances and not only (see next paragraph) – appear only together with concrete material bodies about which one may predicate. But these last are the only things of which one can be sure that Aristotle grants the existence; all the rest (herein the universals) are aspects of the being that can be observed when they are found in (present-in), or recognized by the mind when they are predicated (said-of) of some substance. The form of existence of all this "rest" seems to have no particular interest to Aristotle, so that replying to the above questions about the existence of universals by choosing the *in re* option and claiming that it is the aristotelian answer would be somewhat hasty, since we have no undoubtable clue of Aristotle excluding the last remained position of *post rem*. The only certain point is that Aristotle, at a moment of the development of his system, has conceived any (prime) substance as inseparably made of form and matter, the form being its ultimate essence.

On the contrary, we have more important hints about Aristotle positing the existence of another class of entities, the tropes, or accidental particulars. Indeed, in order to have an accidental universal being said-of something it needs an accidental particular to be there so as to be that about which the universal is predicated – as well as a non-accidental particular (a substance) in which both the "accidents" reside (are present-in). Recalling the example of Socrates (a prime substance) and his knowledge-of-not-knowing (an accidental particular), the accidental universal *knowledge* is such as far as it can be said-of a "thing" like Socrates' knowledge-of-not-knowing. Therefore, the existence of such a necessary "thing" is surely much more convincing, to Aristotle, than that of any universal.

Anyway, we can stick to the simplest case of plain concrete, material bodies and yet note that the form of each (prime) substance is just one, so that we could ask whether it is the sum of all the universals relevant to that substance (everything that can be said-of it while being not present-in it) or, rather, only the universal corresponding to the most proper species of that (prime) substance, e.g. when it is considered as placed within the taxonomy that unfolds the category of substance in the ten-fold schema of *Categories*. With a contemporary terminology we could pose the question whether Aristotle would accept the existence *in re* of a universal for every property of a substance – say for instance in the prime substance named Socrates, the property of being a philosopher, a husband, a Greek and also being ugly, being skilled in rhetorics and so on[7] – or only the existence *in re* of the universal corresponding to its unique essential property (say the humanity, Socrates being an instance of the human species), provided that it is uncontroversial to identify the real essence of any given substance.

The *post rem* position, finally, holds that universals exist inasmuch as they are real products of human mind, produced by means of the process of abstraction from the (non-accidental or accidental) particulars, the things (*res*) after (*post*) the experience of which the universal – the idea of the general property shared by all, basically the underlying *concept* – is drawn. Although there is no singular name in the elder Greek philosophy to which trace back *in toto* this position, its roots can be found again in Aristotle and in particular in the above mentioned observations about the relationship between language and reality. Indeed, already in Aristotle that relationship looks like mediated by some (never better exposed in Aristotle's remained writings) mental experiences. Starting from such a basis, commentators and interpreters of Aristotle had over eight centuries to provide Boethius with such a tripartite classification of the different approaches to the questions on the existence and nature of the universals.

Boethius then attempts also to give his own answer – which by the way most likely relies on arguments by Porphyry whose second, deeper and longer commentary on Aristotle's *Categories* has gone lost, and who in his turn most likely relied on arguments from early Greek commentators on Aristotle. The most surprising point in Boethius' solution is his argumentative process. Indeed, he starts by denying the universals even the dignity to be studied, by means of an argument that basically tributes existence only to singular, quite concrete things (objects that can be reasonably individuated). As a consequence, the universals – which, if they exist, must be shared by many particulars and then cannot be reasonably individuated – cannot even be the objects of an enquiry. But eventually Boethius ends up by holding that the universals, which are the result of a mental process of abstraction from occurring things, are not mere constructions of the mind, rather they grasp the ultimate essence of reality, reality as it really is – it may be enlightening to consider that he took his examples from the domain of geometrical objects such as lines and points. This appears to us as the answer that somehow displaces the deep question forward, asking now *what is reality?*

Though deep and intriguing, Boethius' solution did not end the debate about universals. Quite the contrary, Boethius' Latin translations both of the porphyryan *Isagoge* and of the other writings by the neoplatonic, along with his commentaries, are considered as the basis for the development of that phase of the western philosophy known as *Scholastic* – Boethius acting as the connection between antiquity and middle ages and between the Greek-Ellenistic world and the Latin world. And throughout the whole "golden age of the Scholastic" the problem of universals has stayed central, as well as the metaphysical issues in general – in conjunction with the effort to compose ancients' philosophy with the revelations of the Christian faith. In such a context, two currents

[7] We acknowledge that at least some of these could be taken as tropes rather than universals, but we also consider that such a choice relies upon a previous choice between, in Aristotle's terms, their being just predicable or, rather, not predicable but present-in the (prime) substance, that which reflects pretty opposite approaches to the analysis of substances and, most of all, already requires an answer to the question whether (and which ones of) such properties are essential to the (prime) substance.

have radicalized in a quite hard opposition, the one roughly corresponding to the *ante rem* or *in re* positions and the other simply denying, though with different strength by the case of each single philospher, any existence to the universals. These two currents are known with the names of *realism* and *nominalism* respectively. There is then a third, intermediary option known as *conceptualism*. Note by the way that, as usual, such names have been coined, and tributed to single philosophers and / or to schools, quite later by the hands of subsequent philosophers, whereas no one of the Scholastics would have named himself a realist or a nominalist. Nevertheless, as a result of the long lived debate on such themes, which has pushed to their limits all the possible implications of both options, the choice for one or the other option about the fundamental relationship between language and reality had become a key to distinguish among medieval (Scholastic) philosophers, especially between XI and XIV century.

Realism stays compliant, generally speaking, with the Platonist account of reality. In its "extreme" variant, realism is quite the same as Platonism: universals then are forms independent from concrete material bodies. Rather, these latter depend on the former for their understandability. However, milder forms of realism accept the existence of universals even only within the particulars (a vision closer to the *in re* position). The crucial point resides in the attribution of existence to the universals, which are considered really existing entities – hence the name of realism. In this case the terms in any language which refer to universals actually denote something, some singular entity. And such entities also must occupy some place in the ontology. Prominent figures of realist philosphers in the Scholastic have been William of Champeaux (considered the founder of extreme realism) and Thomas Aquinas, though this last supporting a much milder position.

Nominalism too presents a variety of internal sub-currents – in which sometimes even conceptualism is counted. The basic point of nominalism, as opposite to realism, is precisely the negation of existence of the universals, be it in the things (*in re*) or alone (*ante re*). The positive claim of nominalism is that universals are just names – hence *nominalism*, *via* the Latin word *nomen* for name – i.e. no other entities are to be posited beside concrete bodies (the aristotelian prime substances). That which the terms of a language refer to as universals then are just conventions, being the universals mere expressions of the voice, sounds – which however may enable successful communications. Since nominalism posits lesser entities to be counted in an ontology, it somehow introduces an "economic" principle in ontology – which has been later known as the *Ockham's razor*, after William of Ockham who has been the most prominent figure of the (moderated) nominalism – though, once again, the expression has been coined later, only in XIX century. The Ockham's razor is, roughly, the principle that requires philosophical explanations and scientific theories to be "parsimonious", that is to make use of the least possible number of entities in order to account for any given subject studied. To say it in another way, according to this principle, given two theories both able to explain a given phenomenon, the one which posits lesser elements is to be preferred. The closest formulation of it by Ockham is just "*numquam ponenda est pluralitas sine necessitate*" (never posit plurality without necessity). With respect to the problem of universals, it is apparent that such a criterium leads Ockham to deny the existence of universals as existing entities, as far as he can account for everything by considering only substances. Another prominent nominalist philosopher has been Roscellinus, probably the first to precisely define this position, who however held a much more extreme variant of nominalism than Ockham.

Conceptualism is the intermediary position in between nominalism and realism. It indeed concedes the existence of universals, but only in human mind, thus excluding a form of existence as beings, or substances, in the extramental reality. Thus, while realists may accept the existence of universals either *ante rem* or *in re*, conceptualist philosphers accept them only *post rem*, as the result of the abstraction process. Terms in the language that stand for universals then refer to mental products, that which may appear as the most difficult case to explain. Indeed while the plain reference to extramental entities (like platonic forms), held by realists, relies on the simplest way of explaining the reference mechanism of the language, and while the merely conventional account of the nominalists conceives the language basically as a socially constructed instrument of communication, the conceptualist vision requires to mix the two alternatives and both to face the issue of the ontological status of mental entities and also to extend the functioning of the reference mechanism so as to deal with mental and extramental entities at the same time.

Really, Scholastic philosphers may have not pushed the aspects most strictly related to the language to the point that we have just indicated – in particular concerning the language as a social construction – but nevertheless their arguments about the problem of universals were clearly rooted in the study of the language, to the point that indeed the issue of universals was studied as part of the disciplines of dialectic and grammar – being then not yet constituted the specific discipline of ontology.

Albertus Magnus and his student Thomas Aquinas, finally, are the most prominent responsibles for a compromise solution that accepts all three ways of conceiving universals. Indeed, by framing the whole discourse within the Catholic doctrine, the universals are reasonably *ante rem*, since they are in the mind of God, *in re* since they inform every particular substance, and also *post rem* since our intellect is able to recover almost the same notion of the universal which is in the mind of God by means of the faculty of abstraction, starting from the particular things of which one can have experience.

7.3. The modern shift

With the diffusion of works by philophers and scientists working outside the environment of the Scholastic, the problem of universals becomes less central, though never marginal. It goes somehow in the background, the fundamental question asking now whether it is possible to know anything by means of rational reasoning alone or, on the contrary, it is always needed to base any enquiry on empirical data and experience on them – with a number of intermediary positions. For instance, if we (by means of an explicit and aware oversimplification) stick to the two opposite poles for possible answers to the above fundamental question the names of rationalism and empiricism respectively, we can see, with respect to the problem of the universals, the former ranging over the positions of realism and conceptualism whereas the latter confrontating – sometimes in quite a lively debate – the positions of nominalism and conceptualism.

However, we absolutely cannot here cope with the task of accounting for the development of such and similar debates throughout the centuries up to now. We could not even choose a few philosophers as the most representative of the two main currents, for it would require a guilty insistence in the oversimplification, and whatever selection of philophers would be regrettable because prone to a number of criticisms – mostly for not having chosen someone else. We then just ask the reader to follow our arguments to point out a significant shift in the role of ontology and the study of ontological issues which seems to be the key of the treatment of the matter during the whole modern era. The few philosophers that we will name then are named just because in their thought and in their writings one can identify some passages that make apparent the shift that we aim to indicate, which is basically a shift in the scope and purpose of ontology. Indeed, if we have taken for grant, since Aristotle and up to now, that ontology is that primary science whose task is to identify which things exist in reality, the shift consists precisely in denying that ontology is to do that, basically based on the conviction that it is not able to do that since either it is not a science or its object of enquiry is not objective, i.e. it has not the status to be investigated scientifically.

It is interesting to note then that such a shift takes place just after that ontology has got its own name. As we have mentioned above, the name *ontology* comes from the hands of Lorhard and Goclenius, but it is Christian Wolff (1679–1754) who makes the word widely known. He is still firmly rooted in that which we would call the original setting of ontology, that is the conviction that ontology must unveil what exists, whereas the "rest" of metaphysics should explain what essentially, ultimately is all that exists. And indeed he not only proceeds in his activity according to this conviction, but he also proposes an organization of the sciences in which the place and the role of ontology is well defined in the context of philosophical and scientific branches of enquiry. Wolff envisages an organization of philosophy and science (which were then still considered about the same thing) wherein logic is the central discipline, to which the very first place is due. Nevertheless, although logic is the most important discipline, for it supports the enquiry of any other branch of philosophy and science, ontology stays as the "prime philosophy", as for Aristotle, since even the principles of logic are to be demonstrated, and that which is needed in order to demonstrate them comes from ontology – and also from psychology according to Wolff as far as the rule and principles of human knowledge and cognition are involved in logical demonstrations.

What is most interesting to us, when tackling the organization of theoretical philosophy Wolff largely relies on its presentation by Aristotle and the Scholastics, so that for him (general) metaphysics consists mostly of ontology as it was conceived by Aristotle – the "prime" science that allows to settle all other sciences properly – and here finally named ontology (just consider the book by Wolff *Philosophia prima, sive Ontologia* in

1730). After ontology, other special metaphysical theories deal with specific issues of metaphysical enquiry, such as rational psychology (focussing on the nature of and the other issues regarding the soul), rational cosmology (about the ultimate nature of the bodies) and rational teology (about God and God's attributes). Ontology therefore is the only science capable to explain fundamental notions such as essence, existence, attribute, mode and so on, whereas any other science would not able to do so because these same notions are used in them, so that they cannot be observed with the needed "detachment".

The pretty same organization will hold also for Kant, though under substantially different premises. Immanuel Kant (1724–1804) too, indeed, considered metaphysics as divided into four parts: ontology, rational psychology, rational cosmology and rational teology. And he too recognized to the ontology a primacy, for it deals with the most abstract forms of cognition, purely *a priori*. Later on, Kant admits even that

> How are the *a priori* cognitions possible? The science that answers this question is called critique of pure reason. Trascendental philosophy is the system of all our pure a priori cognitions; customarily it is called *ontology*. Ontology thus deals with things in general, it abstracts from everything particular. It embraces all pure concepts of the understanding and all principles of the understanding or of reason. [Metaphysik L2 (1790–1791) AK 28:tbf, in [Kant, 1997] p. 308]

In this brief excerpt from Kant there are actually all the elements that signal the shift that we are to point out. But in order to properly understand these statements it is necessary to at least briefly recover some fundamental aspect of Kant's philosophy. Focusing just on the position described in his "mature" production – after the first Critique (1781) – we can see how Kant appears as the first after Aristotle to define a system of categories together with, and within, an original and well defined conception of ontology. For the usual limitations (not the right place this work, not the right person this author) we will not examine in detail Kant's philosophy; we must content us with highlighting the consequences that Kant's assumptions bring with respect to ontology.

As in the excerpt, it all starts with the distinction between *a priori* and *a posteriori* judgements. By combining it with the other distinction concerning judgements, the one between analytic and synthetic, Kant identifies beside the analytic *a priori* – which express necessary truths, but are not informative (tautologies) – and synthetic *a posteriori* – which are informative but not necessary and therefore require constant empirical verification – the class of the synthetical *a priori* judgements. In these last Kant finds that which constitutes scientifical knowledge and then he focuses on the possibility of such judgements. All the ontology – in particular the trascendental analytics – is then for Kant the science that deals with the faculty of human intellect to experiment, perceive, understand and finally (really) know reality, but based on the original assumption that reality is not something lying outside, in front of the individual intellect who enquires; rather, reality is the form that intellect gives, by means of reasoning, to the *stimuli* received through senses and interpreted based on the two primary mental functions that are *space* (pertaining to the environment external to the knowing subject) and *time* (pertaining to his internal environment). Indeed, after the "processing" through these functions, according to Kant, the input coming from senses is analysed by the intellect by means of purely *a priori* concepts that resides in human mind. These are the (twelve) categories defined by Kant, which allow to match occurring information from ongoing experience with *a priori* knowledge, and to finally increase this last by expanding the collection of *a priori* knowledge with newly occurred synthetic *a priori* judgements – since these come from experience but are also necessarily valid.

It is important to note here that Kant's categories, though bearing for the most the same names as Aristotle's categories, are absolutely different from those. Indeed, while for Aristotle the ten categories provide a schema for classifying beings according to the modes in which something can be predicated about them – thus assuming that beings do exist in an extramental reality (at least prime substances) about which the language talks – Kant, on the contrary, openly excludes that his categories could be applied to something outside human intellect. Kant's categories are meaningful inasmuch as they are applied to the "mental objects" (which is not an original expression by Kant) which derive from senses and are called *transcendental* since they pertain to *a priori* knowledge (like forms for ancient Greeks) but are meaningful only as far as they are applied to sensible experiences (matter). The coupling of matter and form however does not occur in external reality, but only in the mind of human beings, which actually produce that thanks to the reasoning faculty and a few purely *a priori* concepts, i.e. the categories.

This way, we remark, Kant bypasses the language. Kant's categories indeed, according to his claims, do not affect that which we can predicate about some existing thing – which is on the contrary the position of Aristotle and which, by the way, could be asked whether any person, any culture, any people and any language would accept that selection of categories as insightful and exhaustive. Thus, Kant's categories directly refer to innate concepts that, though implicitly in Kant, must apparently be assumed as common to any human being, since the only universal principle in all the system of the Pure Reason is the "I think", which is nothing but this mechanism of presentation of sensible data to the reasoning faculty which processes them by means of the mental functions and the categories.

Although apparently there is no direct historical connection, the kantian ontology seems to give an answer to the question that had emerged after the presentation of Boethius' solution to the problem of universals: what is reality? And Kant's answer breaks with the tradition that since the time of ancient Greek philosophers assumed an external reality more or less knowable, more or less perfect – as for Plato whose philosophy accounted for even two orders of reality, the imperfect terrestrial world and the perfect Hyperuranium. To be more careful, one clearly cannot say that Kant denies existence of anything outside human reason, but in any case that is not reality. Reality is the result of the activity of the "I think". As a consequence, ontology, though it keeps this name, is no longer actually the science of the being *qua* being; rather it becomes the science that studies the faculty of knowledge, the possibility of knowledge itself, what we today would rather call *epistemology* – or *gnoseology* to preserve the wide perspective of the work by Kant, which indeed was about the faculty of understanding and knowing in its entirety, not only about scientific knowledge, albeit his remarks about the value of synthetic *a priori* judgements concern precisely scientific knowledge.

It would be interesting to consider also the position of René Descartes (1596–1650) concerning this shift from ontology to epistemology. Indeed Descartes, usually considered as the founder of modern philosophy, had already posed the question about the possibility of knowledge – in particular of *true* knowledge – more than a century before Kant. His "methodical doubt" about even the existence of any external reality led him, however, to pretty different conclusions. To put it in a nutshell, and focusing just on what is more strictly relevant to the ontological issue, Descartes still believes in an external reality with respect to which, then, an ontology can yet be thought as the "listing" of existing things, which are given outside human mind, though the possibility to recognize, understand and know what really a thing is becomes problematic. After all, precisely this questioning about the knowledge itself is considered one of the key elements of modern philosophy. Anyway, we are urged here to quickly follow the development of ontology in the occidental philosophy and we cannot linger too long on this point once it is clear that the shift that we intended to point out is this change in perspective and in scope of the ontology: from the enquiry about what exists, in its intimate and ultimate nature, to the study of what and how we can know. This last reading of the fundamental question has sometimes received negative answers, even denying any possibility to reliably access, understand and know reality.

With extreme simplification, we just sketch here below the basic arguments presented by the main philosophical currents that, after Kant and up to the next important turn that we aim to signal, either have given a negative answer to this question about knowability and the scope of knowledge or have nevertheless put in the background that which had been at the very core of philosophy since Parmenides.

After Kant, occidental philosophy goes through the season of idealism, wherein some aspects of Kantian philosophy are brought to extreme conclusions, so that the principle of the thinking reason becomes the origin and the cause of everything and there is no room, nor any interest, for external reality: reality is considered to be created by the mind, the soul or the spirit according to the way how single idealists present their own position. Being there no place for external reality, the idealism does not need a science to study existing object other than the introspective observation which allow to contemplate and possibly study the reality as it is produced by the mind.

Not even the most critical answer to idealism – that is materialism – reassigns its role to ontology, though Marx, for instance, bases all reality on strictly material foundations. Indeed, on such a material base any other social, cultural, non-material "entity" is constructed, and also determined by. Nevertheless, this does not imply the development of a strictly materialistic ontology that acounts for the basic relationship involving language, mind and reality. Rather, in marxist materialism the fundamental relationship involves the man (as the mankind) and the nature, connected by means of the productive activity of work. Reality then is interesting as far as it is the counterpart for the life of human beings, and the distance that divides man and nature is a measure of the quality of life and eventually happiness of a man. The ontological status of things and beings and their constitutive relations are not interesting *per se*, but they become important when, once clarified, they serve to ground and defend ambitious projects of social justice aimed to recover the right configuration in the fundamental relationship between man and nature, reality.

Yet other philosophical currents have appeared in "late modernity" and in contemporary (western) philosophy. Most of them had no place for ontology intended as the discipline or the branch of philosophical enquiry that deals with the "inventory" of existing thing, that is tightly coupled with metaphysical issues concerning the ultimate nature of things and that, as a consequence, must cope with the relationship between language and reality so as to discriminate between what actually pertains to the objects (extramental things, the beings *qua* being) and what instead depends on our modes of knowledge about reality, a knowledge that in any case is managed mostly by means of the language. Quite the contrary, such currents usually feel no need of ontology in this sense, for a number of different reasons. Just to be informative, but not exhaustive, in what follows we just try to point out, by means of a drastic simplification, the fundamental assumptions taken by some of these currents that basically account for their refuse of ontology – and clearly we consider here just the assumption most relevant to the ontological issue, leaving aside all the other aspects of the thought of many authors working in each of these currents.

Nihilism, as an extreme development of idealism at least on this respect, definitely excludes the possibility to reach for an understanding of reality which may claim any reliability with respect to an external, substantial reality which is not only in the knowing subject's mind. Thus nihilism finally rejects even the possibility to take seriously into consideration external reality: human beings can just perceive the appearences of phenomena and any further consideration based on these can lead just to a discourse confined within the single mind, with no possibility to scale up to have (certain) knowledge about outer substantial reality. Therefore, nihilisim needs no ontology.

Positivism on the contrary brings great faith in the possibility to understand and know substantial reality, beyond the knowledge of what is perceived. But the "positive faith" is rooted in the progress of positive sciences, which especially in XIX century made many thinkers and philosophers get convinced that the ultimate understanding of nature and the whole reality were ready to be catched precisely by means of positive sciences. Metaphysics then, and ontology as its part, appeared to positivist absolutely useless, since not scientific according to the criteria of positive sciences. However, one should note that positive sciences and metaphysics move from quite different fundamental questions: *what exists*? and *what is the ultimate nature of that which exist?* for the latter and *why do things happen?* and *how do they happen?* for the former. And also move toward quite different objectives: metaphysics tends to global unitary ultimate account of everything, whereas positive sciences tend to detailed modelling of pieces of reality, possibly producing theories compatible with each other.

Logical positivism, or neopositivism, emerges at the beginning of XX century and lasts a few decades, reaching quite soon the dead-ends of its assumptions. Though not so tightly connected with the "previous" positivism as its name could suggest, it appears nevertheless as the most rigid form of positivism. It can be roughly defined based on its rejection of metaphysics in general and ontology in particular, together with the ambition to construct a purely logical language by means of which – essentially for its formal aspects – almost automatically get rid of all problems in philosophy and in sciences, thanks to the purity of this (chimeric) language whose referential mechanism maps any term and any proposition exactly – i.e. with no room for ambiguity – into objective references (objects and states of affairs respectively) in reality. The reasons of the reject of metaphysics and most of all ontology lie in one fundamental assumption of neopositivism concerning the kinds of judgements that the science is made of: the class of *synthetic a priori* judgements – that Kant had found as those of which scientific knowledge is mostly made of – is deemed senseless by positivists, who only admit the neat distinction between analytical (*a priori*) and synthetical (*a posteriori*) judgements (propositions). In particular, whereas analytical ones are considered sufficient to provide a complete account of mathematics, all the other sciences must rely only on synthetic *a posteriori* which, as Kant noted, always need verification. Positivisms indeed is really rooted in the principle of *verification* that requires precisely that for every "cognitively meaningful" proposition there must be a finite procedure, perhaps one could say an algorithm, to verify it. The point therefore is no longer in a difference of purpose and / or leading questions between ontology and the other sciences – as it was with respect with "plain" positivism – but in the complete disregard, on the part of positivism, for ontology as a discipline that may say anything meaningful, if not changing to the register of artistic, poetic, expressive literature; and on the other hand the absorption of all sciences under the epistemological umbrella of verificationism, which should have lead to the unification of all sciences on the basis of a unique scientific language.

Existentialism, on the contrary, moves from the "right questions", but it suffers the *impasse* of being not able to give them any answer. Existentialist philosophers then focus mostly on the "existential questions" about the specificity of the human being (such as *what is the sense of life?* or *for what do we exist?* just to suggest the way how questions get turned) moving from the acknowledgement of the impossibility to cope with the fundamental questions concerning existence and the being in themselves, to the focusing of fundamental questions on the human condition, looking for what makes (human) existence meaningful, and a human life "well lived". They also move in the distinction between beings (given entities, existing things, aristotelian substances we might say) and the Being intended as that which is unveiled to the knowing subject after the encounter with single, particular beings. This latter Being is recognized as the only interesting object of philosophical enquiry, liable to be investigated about the possibility itself to be objectively known – which is in general denied in existentialism, since knowledge of it depends on the particular series of encounters of an individual with the Being that she meets time by time during her existence.

Finally we could signal also the position of deconstructionism, emerged only in the second half of last century, even though it is quite far from being a philosophical current. Anyway, as an approach to philosophical issues, it complaints the bag of troubles and aporias that every philosophical system bring with itself, but also the inescapability of metaphysics, usually considered as the source of such troubles. We just mention it here in order to count also this last position in our (yet incomplete) survey of the positions critical, contrary or even hostile to ontology – as part of metaphysical enquiry – so as to fill in the scenery with all the necessary elements with respect to which we will try to make emerge, in the rest of this work, the points that, we believe, make ontology not only a discipline worth of study with the same dignity of other sciences – respecting the differences in the purpose and the specific questions that are faced – but also a fascinating challenge in the special context offered by the application of ontological research instruments and methods to the virtual world of information systems and, still more, of World Wide Web.

7.4. The formal turn

Out of this chorus is the position of phenomenology, formulated by Edmund Husserl at the beginning of XX century. To be more precise, we mean in particular the "realist phenomenology" since later "trascendental phenomenology" leads Husserl (and the others following him in this line) to positions very close to that of Kant, which are also the starting point for many existentialists after Heidegger's interpretation and use of phenomenology.

Nevertheless, in Husserl's phenomenology appear a number of elements which turn out to be central for our outline of the development of ontology and of the ontological conceptions in occidental philosophy up to today's use of ontology in information systems. First of all, *Logical Investigations* – the work wherein Husserl introduces the (realist) phenomenology – mark a return to the objects, given and present (in some way) in front of the subject who observes, understands and knows – that which basically makes room for an ontology comparable with the conceptions before Kant. Secondly, is recognized to ontology a proper, specific and formal method that, along with recognized and well defined specific purposes and scope, preserves the role and autonomy of ontology as an independent discipline. It is the central point to us, that which causes the *formal turn* in ontology. And thirdly, the process of understanding and attaining knowledge is explained by the analysis of the *representations* that occur in the mind of the observer – that which makes explicit in our discourse, though this issue was implicitly already recognizeable at least in Kant, the distinction between the two aspects of (external) reality and the representation of it that builds up our knowledge.

Actually, Husserl's work is originally moved by the intent to refute and confute the positions according to which the analytical *a priori* validity of the laws of logic could be demonstrated on the basis of psychology. For this reason Husserl introduces phenomenology as a method to attain the primitive stage of knowledge, i.e. before that ordinary conventional "constructions" take place in the process of understanding and, thus, bias knowledge. Husserl then envisages phenomenology as the way to affirm philosophy as a scientific discipline with its own object of study in perception and mental representations, but from a completely different point of view with respect to psychology. As a consequence, and coming back to our interest in ontology, in order to know what a thing is, Husserl applies phenomenology. In its first version, it can be compared to a realist position about ontology. It is only later on that Husserl chooses to eliminate any assumption about the existence of the objects, the external references of conscious representations. Such a choice makes literature talk about the change of Husserl towards trascendental phenomenology in lieu of the previous, "realist" phenomenology. And also causes Husserl, due to the epistemological issues that supersede ontological ones, to collapse ontology and phenomenology, "bracketing" the external world and focusing only on the reality that is built by means of mental representations.

To better illustrate these aspects, we need to introduce at least the basic elements of phenomenology. We will introduce them in a very simplified way – yet not trivializing it – just rich enough to let emerge the above mentioned aspects. It considers the way how a subject may know an object, so that the very first, and the most crucial to Husserl, aspect is the special relation holding between these two. The mature setting of the question considers the activity of the subject as the process by which the object is observed from as a neutral as possible point of view, in a sort of detachment from ordinary relation with everyday reality, whose perception is biased by many theoretical or common sense convictions, so as to exclude prejudicial interpretations. Once "put" the object in the correct relation, the subject can properly perceive it in a two-steps process that passes through the *sensible intuition*, which operates on data from sensory perceptions and produces a mental representation, and the *categorial intuition*, by means of which consciousness applies to the mental representation of the object the relevant formal categories. This second step of the process is precisely that which offers the most important hints about ontology. Husserl indeed distinguishes between two kinds of *formal categories* that apply to representations, respectively for meanings and objects. On the other and, concerning objects, he distinguishes between formal categories and material categories, which deserve very different treatments. Let's focus for a moment on these different kinds of categories.

Meaning categories are built based on the observation of language mechanisms and allow to identify classes of meanings – that are correlated to corresponding classes of objects – by means of a sort of syntactical substitution test that works by highlighting nonsenses. For instance, in the sentence "The table is round" the word *table* cannot be substituted with other words like *red* or *beautiful* unless obtaining a nonsense, which is enough to indicate that their meanings belong to different meaning categories. It is easy to note that the idea of meaning underlying this mechanism is rather a matter of syntax, so that meaning categories could be roughly assimilated to grammatical categories and the above example could be explained based on the different categories of nominative expressions (like *table*) and adjectival expressions (like *red* or *beautiful*), which are not interchangeable in a sentence like that. It is also easy to note that such a correspondence yields to the correspondence with objects: nominative expressions are, roughly speaking, objects as well as adjectival expressions are properties, and sentences correspond to states of affairs. After all, we have said that Husserl poses the objects of meaning categories and those of formal ontological categories as correlated.

Formal ontological categories on the contrary reflect objects and their formal properties, that which could even look not so different from the previous meaning categories. Indeed, the criterion to find and justify formal ontological categories seems to be again that of nonsense tests – whereas an empirical approach that considers what actually objects are will pertain just to the family of material categories. Thus, for instance, a "round and not-round table" will find its place in no formal ontological category since it is formally absurd. The formal character of such a consideration lies in attributing to a given object both a property and the property of not having that property, that which causes a formal inconsistency, independently of what actually is that property. Although in this particular example it may seem that formality too could be exchanged for syntactical fiddling on the sentences that carry – and perhaps also constitute one should ask – some representation, the treatment of formal ontological categories really claims for many aspects much more adherent to formal properties of objects in general, concerning e.g. the relationship between an object and its parts, or the relations holding between objects – like also in the example the point is in the relations holding between an object and a property. Formal ontological categories, then, have their subject matter in concepts like *object*, *unit*, *state of affairs*, *plurality*, *relation* and so on. These indeed are the proper objects of investigation of the discipline christened *formal ontology* by Husserl.

Material ontological categories finally are the ones devoted to that which one would typically expect as the subject matter of ontology, if ontology is understood as the listing of existing things. Material categories then are built based on the nature of the objects and actually seems to be not so important to Husserl, who rather insists on formal ontology, whereas material categories are the result of a slightly different ontological activity, that is given no special name but delivers quite different results such as *regional ontologies*. The scope of this last is not universal, as it is that of formal ontology, but limited to the particular region that is the field of interest (current terminology would say *domain*). Yet another kind of nonsense or absurdity test may lead to the discovery of material ontological categories, like for instance in the expression "A round table", the word *table* cannot be substituted, unless obtaining an absurdity, by words such as *triangle* and *morning*, though they all are nominative expressions, for the absurdity depends on some aspects of the nature of the objects correlated to these terms.

On the one hand then, the first distinction, between meaning and ontological categories, somehow makes explicit the mechanism that was at the basis also of the system of aristotelian categories, as the word *category* itself comes from the Greek word for *predication* and, most of all, the separability of the category of (prime) substance from all the other show. In a similar way indeed, Husserl's meaning categories allow to separate that which can be said about objects from objects themselves working on the level of representation itself, i.e. the language, whereas ontological categories distinguish kinds of objects based on their formal or on their substantial, essential, natural properties. On the other hand the second distinction, between formal ontological and material ontological categories introduces the major novelty of Husserl's reflection on ontology, and systems of catgeories, by distinguishing two alternative and independent methods to perform ontological investigation, one operating by means of formalization, abstracting from the particularity of the objects up to the "erasure" of their essential nature to keep only the form involved in the relationships with other such forms, and the other pretty traditionally focusing on the very essence of the objects. Moreover, these two methods, and the resulting taxonomies – one immutable, in principle, formal ontology, and a collection of regional ontologies – together compose a rich and flexible array of categories for classification.

The most interesting nevertheless is the formal ontology, for which we speak of a formal turn of ontology. Formal ontology is still about categories of objects, but these are to be investigated and produced purely *a priori*, since Husserl's aim is to develop the discipline whose purpose is not to produce the inventory of existing things – especially after the trascendental variant of phenomenology – but rather to study the properties of objects and their relations at the most abstract level. Formalization, after all, means for Husserl emptying of content. Progressive abstraction is also the structuring relation of the taxonomy in which the categories of formal ontology should be organized, so that the highest genus should be something like "object in general" – which is the most abstract case of object. On the other hand we may note also that, according to Husserl, the definition of more "traditional" categories, capable to classify the existing objects based on their material properties and their nature as existing things, pertains to quite a different ontological practice, that of regional ontologies. Material categories, then, are organized based on another criterion, that we could call *generalization*, something that spots common aspects of the objects of a given region, so that the highest genus should be the largest object to which correspond the weakest concept – something like a "least common denominator" of the concepts (and objects) of that region.

So, whereas aristotelian categories were based on predication (in such a way that Aristotle's categories, used to classify objects, are derived from the language and reflect different modes of predication) and whereas kantian categories are based on thought (that is correspond to the maximally general conceptual scheme according to which we understand reality, but it does not regard necessarily external reality) Husserl's categories operates somehow separately on the three levels of language (meaning categories), thought (formal ontological categories) and material reality (material ontological categories). Moreover, whereas the meaning categories and the formal ontological categories share the formal approach (and offer hints for the identification of categories for each other, so that formal aspects of objects can be highlighted by language mechanisms and *viceversa*), the material ontological categories follow the alternative way of more empirical research, thus providing a very rich system for classification.

In fact, we could say that regional ontologies are to formal ontology as special theories are to logic. That is, like logic studies formally the propositions and their relations and in particular how the truth of a proposition may lead to the truth of another proposition independently of their meaning, formal ontology deals with objects and their formal relations independently of their particular nature. It is from this line of research, by the way, that emerges the study of mereology, as the specialized study of the formal relations between an object and its parts.

As mentioned above, later development of phenomenology by Husserl himself could be interpreted as leading to the acceptation of "non-existing objects" as contents of consciousness, since subject may conceive an object even without having been presented with it. That is, categorial intuition may take place and elaborate some representation even without the previous production of the representation based on a sensible intuition, in the absence of an object given in front of the subject. It is perfectly compliant with the assumption about a processing of formal-ontological enquiry which is totally *a priori* and focused on purely formal properties of objects, which therefore cannot be but mental objects hosted in mental representations. The initial return to the objects of an external reality, then, is traded by Husserl for a purely formal study of reality that must leave aside any particular given property of an object, such as existence itself – this is also the reason for the detachment of formal ontology and regional ontologies. In this way, the proper place where objects can be formally studied is in the mental representations. Now, although Husserl does not limit the instruments of formal ontology to language analysis – rather, he seems to be inspired by the novelties of his time concerning logics and mathematics – nevertheless the choice of working on mental representations, since one can analyse them customarily by means of the language, opens a new chapter of the issue about the relation between language and reality, and the mind as the place of representations. However, we will get much deeper in the details of such an argument in next chapter, where all the hints and interesting points emerged here will be matched against other hints emerging from the study of the representations of knowledge by means of Ontological Compatibilty Spaces, that is where a sub- or pre-linguistic, geometric point of view can be reached.

7.5. Formal ontology, formalized ontologies and formalism

We have somewhat insisted on *formal ontology* because indeed it seems to us really central to all subsequent developments of ontology since Husserl's time. In particular, we are going to focus on the term *formal* and based on it we will identify two main lines of development of ontological enquiry along XX century and up to now, taking some inspiration from the argumentation of Poli ([Poli, 1992, 2003]).

The first line, so to say, is philologically more careful and insists on the formal aspects of ontology as they had been pointed out by Husserl, that is essentially as an analysis of formal ontological categories like *object*, *part*, *whole*, *number* and the like. It is the acceptation of *formal ontology* that is taken mostly by ontologists which are philosophers. Within this orientation of ontological enquiry, all problems of ontology (like typically the problem of universals) have been recovered and investigated on the basis of a much more robust framework, thanks to the rigor of formal analysis. Actually, any philosopher involved with ontology today has to face a system of formal categories. This does not mean that it is the last word on the subject nor that everyone directly confrontates with Husserl's system, but the method and the criteria of his formal ontology, appear as an inescapable point to any philosophical work on ontology that aims to provide a robust system of categories. In this sense the formal turn of ontology seems to us a capital passage in the history of ontology, to be signalled together with the first attempt to a system of categories by Aristotle and the modern opening, with Descartes and Kant in particular, to epistemological aspects.

The second line moves, so to say, from a sort of biased interpretation of the term *formal* itself, which suffers a shift of its meaning from pertaining to formality to formalism. That is, it suffers the change of focus from the study of what remains after (extreme) abstraction from the specific content of the objects of enquiry, to the study of what the abstractions are represented and manipulated by, which turns out to be most of all the study of the syntax and syntactical aspects of some special purpose language, i.e. some formalism. After all, one may note that this "pathology" threatens also formal logic. In fact, as Poli reports ([Poli, 1992]), formal ontology has been conceived by Husserl in tight correspondence with formal logic. And formal ontology may explain all its significance through the comparison not only with material ontology, but also with formal logic. With material ontology it shares the purpose and the object of study, but these two distinct "disciplines" of ontology adopt different method and approach. With formal logic on the contrary the comparison is suggested precisely to indicate a sort of model whose approach and method should be followed or imitated, though for a different purpose and with different objects of study. The possibility to shift from the study of formal, abstract structures by means of their representations in some language that makes them liable to be analysed, to the study of that language itself – as just one among many other possible languages – and of its syntactical properties is therefore quite close for both formal logic and formal ontology.

Nevertheless this second line of development of ontology, that we would better call and refer to as *formalized ontology* (rather than formal), much more than being a misunderstanding of *formal ontology* has its own significance as a sort of applied science, or better as the "porting" of ontology in the field of research in computer science in general, and information systems and Artificial Intelligence (AI) in particular. It is quite a different issue also with respect to *applied ontology*, since this last concerns also many other domains, such as laws, biology, health just to say a few, and in these cases one would rather speak of regional ontologies, pertaining to the distinction introduced by Husserl between formal ontology and material ontology – which could all the same be produced as a collection of regional formalized ontologies. The point with formalized ontologies, then, is precisely the centrality given to the formalistic aspect of the language, intended as a precise, specific language adopted in a particular system, be it a research project for a single intelligent agent in AI or a network of common computers enabling a variety of information services, as it is the case with World Wide Web and the formalized ontologies of Semantic Web. In this case the focus on the formalism is intended to make a set of ontological categories accessible and usable by machines (material or virtual) so as to have them performing operations and task for which a certain amount of knowledge about the world is required. Typically it is just a small fragment – the so-called domain of interest – of human knowledge, but in principle it could be even all human knowledge. Or at least all human knwoledge that can be reduced and expressed effectively by means of some formalism.

Formalized ontology then is the artefact that results from such an effort to decompose and reassemble human knowledge about the world for the "advantage" of a mechanical intelligence – although the final benefit is for the human beings that have machines acting maybe in a virtual dimension, but producing results that affect real world. In this sense we may call such formalized ontology also *computational ontology*. One might argue that it looks like a sort of second-hand ontology, since it is not ontology for the sake of ontology. Nevertheless, the effort to produce formalized regional ontologies for many branches of human activities has sometimes given impulse to fundamental ontological research for a better, deeper understanding of those branches. It is somehow an indication of the accuracy desired for these artefacts the intellectual frequentation that many computational ontologists involved with important projects of formalization of rich regional ontologies – or even foundational ontologies, which actually correspond to the idea of formal ontology itself to be formalized – even though they are not all philosophers, have with the (traditional) ontologists currently most active. Philosophers inclined to contamination with applications and in particular with the formalization of ontology for use within information computing systems are to be counted mostly in the "party" of analytic philosophy, which appears also as the current most involved with ontological enquiry after the refusal of metaphysics sanctioned by "continental" philosophy, in many of its currents as presented above, during last century. That which, by the way, sounds quite curious since the typical (though rough, inaccurate and oversimplifying) distinction between analytic and continental usually identifies the roots of the two streams respectively in neopositivism, which refused ontology together with all metaphysics, and Husserl, who first introduced formal ontology. Nowadays, on the contrary, formal ontology looks like the privilege of thinkers in the analytic tradition.

Nevertheless even in the Old World – the mainland of continental philosophy – the firm opposition to metaphysics is getting challenged by philosophers[8] who see not only the need to recover a formal ontological research, but also the actual impossibility to leave aside metaphysics while progressing with (other) sciences.

From another point of view, however, the development of formal ontology in the tradition of analytic philosophy appears absolutely obvious. Indeed, once acknowledged the inadequacy of the neopositivist programme – due in particular to the wreckage of the attempt to define a "perfect" language, capable of eliminating any problem concerning the reference issue (e.g. naming entities which do not exist) and whose aim was that of formalizing about every scientific theory, in such a way that all authentic, valuable knowledge could have been shown and proved simply within that language – that which remained to analytic philosophy is an approach to philosophical enquiry marked by a method inspired by formality and rigor that typically attempts to circumscribe and define specific problems by means of formal logic, with much more reasonable expectations.

We have not the space here, nor the intention, to embark on a detailed discussion of the intercourses of analytic philosophy with ontology. It is neither in the lines of this work, whose primary interest is in computational ontologies and how to use them within the World Wide Web. Nevertheless we must at least quickly note that the tendency to resort to formal logic to tackle specific issues – which are mostly concerned with the problems of language and knowledge – has made analytic philosophy to be the most natural interlocutor for people concerned with formalization of knowledge in "artificial brains" at the time of AI and still today as far as large part of the results (and troubles) of classical AI (based on explicit symbolic representations of knowledge) are reframed in the context of World Wide Web.

[8] Such as, to name just one for all, the Frenchman Nef who, for instance in [Nef, 2009], shows that no science nor other kind of discipline has ever been able to really leave aside metaphysics in general, and ontology in particular, in setting and performing its specific enquiries.

Therefore, we focus on yet another moment of the history of ontology that has remarkable consequences on computational ontologies, that is the introduction of the notion of *ontological commitment* by Quine. Ontological commitment expresses the requests that a language demands on reality to accomodate the meaning of its sentences – assuming that these sentences are describing some portion of reality – talks about chimerae and unicorns are not to be taken under the lense of analysis for ontological commitment. We have already met the notion of ontological commitment as it is used by Guarino to explain what precisely are computational ontologies (in the first part of this work, p. 54), and we will not insist too long on how it "works". We simply note that the ontological commitment emerges in its full extent only after accurate formalization in first order logic, as Quine puts it. And since formalization is not such a trivial process it has been conceived for use (only) with scientific theories in order to assess their actual implications on reality. By means of such an analysis conducted on the whole scientific knowledge the analytic philosopher would obtain all the elements necessary to deploy an ontology (as a complete system of categories of beings) that is endowed with the formal rigor of logical analysis and the empirical evidence of sciences, provided that this philosopher is willing to change the ontology according to corrections and adjustments of theories to comply with new discoveries presented by scientists. In case of alternative, competing theories on the same subject, the analysis of the ontological commitment is the necessary prerequisite to proficiently apply Ockham's razor, so that among many possible theories able to explain a given phenomenon, it is correct to prefer the one that commits on the fewest as possible entities. Nevertheless, it may not be univocal how to formalize a given proposition: as linguists well know, some sentences can be formalized in more than one way, thus implying different meaning. Similarly, and with implications by far more important, when attempting to formalize a scientific theory the analytic philosopher might encounter interpretation issues that can be solved only by choosing between alternative possible interpretations, that lead to different ontological commitments. After all, as the wreckage of logical positivism has shown, logical formalization is not a magic recipe to solve, all alone, any problem – neither it will be the case for the definition of a univocal ontology.

It is interesting to note then the peculiar relation that holds between this approach to (formal) ontology (for the sake of ontology) and the role assigned to computational ontologies by Guarino through his definition which indeed relies on the notion of ontological commitment. The former evaluates the ontological commitment of scientific theories by means of the logical analysis of the language in which they are expressed, trying to purge scientific propositions of faulty interpretations dependent on the vagueness and inaccuracy of the linguistic expression, with the aim to produce an ontology as a theory of what exists. The latter establishes the ontological commitment of a language by defining a theory (the computational ontology) whose logical interpretation – its formal semantics – is intended to provide the ground and its boundaries for the interpretation of the language, for an artificial language designed to enable machines to share and exhcange knowledge. In other words, the former produces an understanding of reality through the formal analysis of the language; the latter produces a (somewhat fictional) formal reality in order to have an understandable language.

Our interest with all this, therefore, is the light that the notion of ontological commitment sheds on the relationship between reality and language, by means of the formal study of the representations of the former by the latter through logical analysis. The logic to be adopted in order to properly formalize scientific theories, according to Quine, is First Order Logic. This last typically relies for its interpretation on set theory, think for instance of the support set from which are taken the values for the variables in the formulas that compose a first order theory. Quine himself considered sets as fundamental components of ontology since they are necessary to assess the ontological commitment of a theory, and in this sense every theory needs sets (and some set theory)[9]. The depth of such an involvment of set theory with ontology is testified by Quine's slogan "To be is to be the value of a bounded variable" ([Quine, 1948]).

It seems quite an innocent assumption, at least as far as it is accepted as the way to highlight the types of entities which are really needed to ground a theory onto reality, by tying its language to the actual implications that it brings about reality. In this way indeed Quine can distinguish between language constructions and actual references – as Russell did before Quine by means of his theory of descriptions – and points out what is actually to be acknowledged as existing in reality. However, once identified the types of must-exist things, the presence of sets, in the sense of the objects of a formal theory of sets corresponding to these types, becomes somewhat cumbersome since sets have their own peculiarities which do not fit with ontology, like for instance their being out of space and time, immutable and eternal.

This point appears intriguing also with respect to formalized ontologies, and in particular with computational ontologies for informative systems such as the ones conceived for Semantic Web applications. There indeed we find the (engineering) solution consisting of the sharp distinction between the intensional value of concepts and the extensional "matter" of corresponding sets, that which allows to do different operations at the different levels, but leaves open the same theoretical questions concerning space and time, declined in the specific issues of different "filling materials" for the same concept in different places (one may read data sources) or the same extensional matter for different concept names (that is the problem of the location of the information over the Web, which marries with translation problems and ends up in the ontology mapping troubles), and of elements' turn-over within sets (that is mutability of sets over time).

Such issues concerning computational ontology emerge now with new force with respect to past experiences of AI, though most formalisms and techniques (like the separation of intensional T-boxes and extensional A-boxes in knowledge bases) come from that period of intense intercourses between computer science and philosophy. At that time however the aim and the ambition were quite different. In particular, for the current of "classical AI" – the one interested in high level, abstract or conceptual intelligence and logico-formal reasoning, as opposed to earlier cybernetics and later robotics, the aim was to provide a virtual intelligent agent – customarily software running on a single computer with no exchange with an external world – with the symbolic representation of a portion as large as possible of human knowledge, or as

[9]Quine himself by the way provided an original theory for sets, namely New Foundations (NF set theory).

detailed as possible in case of projects focused on some specific set of intellectual tasks. Uniqueness of the intelligent agent and its seclusive isolation made it enough for researchers and the agent to "agree" on one language to be used for input-output delivery and the system was able to perform that for which it had been purposely programmed in order to show an intelligent behaviour. The production of an ontology (though by that time it may have not been called like this), by means of which to teach the agent about its fictitious world (typically very close to a kindergarten, rich in coloured bricks), was aimed precisely to *build* the reality in which the agent had to operate. The language used then had to be formal (a formalisms) to be handled by a computer, and its relationship with the reality was absolutely constitutive, since the only reality was that of the little colourful world of bricks – though some other (sadder) intelligent agents were put in a world made of, say, the axiomatization of mathematics by Russell and Whitehead and lived just to infer mathematical theorems.

Once get out of the seclusion, nowadays intelligent agents for the World Wide Web – which to be honest are yet quite far to appear if one does not content with *ad hoc* single-purpose *web bots* – should be able to collaborate, share and exchange knowledge with each other. Basically, they should communicate. Possibly meaningfully and beyond the strict observation of a predefined protocol that previously establishes which infomation can be dealt with and succesfully transferred. The computational ontologies, then, should be to these agents not only that which produces their limited and simplified reality. Rather, as Guarino puts it while giving the definition of (information systems) ontologies, the ontology should ground the artificial language of the agents onto some reality in order to enable them to effectively communicate. And since these agents are supposed to use the information available in the Web in order to perform annoying tasks on behalf of their human masters, the reality described by the ontology, to which the language of the agents commits, should be also compatible with the reality of the human masters. Briefly, computational ontology for the open Web is by far a more complex work than building artificial intelligent systems. And it deeply involves philosophical ontology.

We think that it may be useful to introduce another criterion to distinguish the different possible purposes and ways of doing ontology. It is not really an emergence from recent trends in ontology, since it can be effectively applied even to ancient Greeks philosophers; but it proves very effective with today's panorama of ontology as it allows to clearly contrast for instance reductionist with realist positions. It identifies two "modes" of ontology: the *descriptive* and the *revisionist* one (cf. Mulligan [Mulligan, 2000] or Smith[10] [Smith, 2003] for instance). To put it in a nutshell, the descriptive ontology tries to stay close to common sense, to intuitive understanding of reality. Typically it provides an account for entities at any level of analysis (macroscopic, mesoscopic and microscopic). For instance, it may contemplate star systems as well as Ozone molecules and broad-leaved trees. Moreover, descriptive ontology typically goes with explanations of phenomena that equally can set the problem at the most convenient level of observation. Briefly, descriptive ontology tends to be close to the

[10]Though actually Smith speaks of *descriptiveness* and *generativity* of ontologies, each one setting its own trade-off between generativity and descriptiveness.

apparent image of the world. On the contrary, revisionist ontology does not care for the apparent image of the world nor for common sense, and typically tends to *reduce* entities and phenomena to their fundamental components, searching for a minimal set of elements sufficient to account for all reality thanks to a special theory of the combinations and aggregations of the minimal elements – in this sense it may be called also generative ontology.[11] One may say that descriptive ontologies are typically larger (as regards the number of entities accepted as existing) and support explanations somewhat redundant – since the same phenomenon may be faced from different persepctives at different levels of observation. On the contrary, revisionist ontologies are typically smaller in the number of the admitted entities, offer only one way to explain what happens in reality and such explanations may easily and quickly reach a terrible degree of complexity.

Now, it is interesting to consider this two modes of ontology with respect to computational ontologies. If considered from the point of view of classical AI or, more generally, framed in the context of an autonomous artificial intelligent agent, a computational ontology should be absolutely generative. It indeed has "only" to produce the reality wherein the agent will act, and such an ontology will do that quite concreteley: in the virtual world made of bits and bytes of a software program, a formal statement asserting that there is such and such a configuration of bricks makes them actually appear in the configuration state of the program – and perhaps one could even find their really physical correlates in some RAM addresses in the computer.

But if we consider computational ontologies in the open context of the World Wide Web and look at them as the artefacts containing the knowledge representations that should enable effective communication between intelligent agents, we need descriptive ontologies. Of course, we must care for the exclusion from this picture of that form of communication conceived as an exchange of typed data according to a (single) predefined schema, which would mean *ad hoc* programming. In this case it would even be presumputuos to speak of ontologies, and XML schemas would do the job, with no need for a choice between (realist) descriptive or (reductionist) generative ontologies. If, rather, we want to conceive, at least in principle, web agents sharing knowledge even in the (most probable) case where they do not rely on the same ontology, then we have to deal with descriptive ontologies, for a number of reasons. First of all, a pretty pragmatic point: people involved with Semantic Web are not all shrewd philosophers or keen ontologists – quite the contrary – so that many (most) available ontologies are just pretty good descriptions of human scale objects and processes, seen from the common-sense point of view, suitable to appear in intuitive explanations of, say, business processes, industrial activities, welfare policies . . . , with no real ontological intent.

Moreover, there is the issue that we have already considered above, concerning the compatibility between human reality and the reality purposely built by knowledge engineers for web agents, so that these last may operate in it accomplishing real tasks on behalf of their masters, with consequences that stripe in material, human reality –

[11] It is easy to see how these categories apply also to ancient Greek philosopher, Aristotle's system of categories being clearly an example of descriptive ontology and, on the other hand, Democritus' and Leucippus' atomism being a revisionist (reductionist) ontology.

think for instance of the booking of a round way flight and hotel stay for attending to a conference on the other face of the planet. For this simple and fundamental reason, most of all, it needs that ontologies are good representations of that portions of reality that will be affected by the actions performed by agents. It is desirable that such representations reflect, descriptively, as close as possible an intuitevely understandable, yet in any case truthful, theory concerning the objects, processes and facts involved with those activities, since the human users of such enhanced services should always keep control on what is happening, and possibly get a precise account of any step of whichever transaction performed by their agents.

The use of descriptive ontologies, however, looks convenient also from a more theoretical point of view that we consider really interesting. The sharing and exchange of knowledge between web agents can be considered generally as a problem of communication. Now, we have already met in the first part of this work the problem of the indeterminacy of translation (cf. p. 78), a thesis clearly posed by Quine. Briefly, this thesis signals the great importance of the information about pragmatic aspects in any communicative exchange. The limit situation from where Quine's argument emerges is that of a field linguist who tries to "decode" a hitherto unknown language by interacting with native speakers of that language. This situtation, *mutatis mutandis*, is not so far from the one where a web agent, with its collection of "known" ontologies, stumbles upon a data source that "speaks" another language. Indeed, despite the lack of large and important parts of human intelligence and cognition capabilities, which surely would come into play in the case of a field linguist discoverying a new language – capabilities that would greatly help him in understanding what native speakers are saying – the case of web agents is even simpler: there are standard languages, the ones provided by the World Wide Web Consortium, that allow to clearly identify and distinguish, say, a concept from a relation, and also quite robust semantics that allows to preserve the logical meaning of the terms used in every ontology. Nevertheless, an agent will still miss the actual meaning of the terms appearing in a new ontology for which it has not been given mapping instructions.

In such a situation then, all that a web agent can see, perceive and grasp of the pragmatics that goes along with the communicative linguistic (syntactical) exchange, is limited to the resources, accessible over the Web, which are labeled with the terms appearing in the new, unknown ontology. But then, also the problem of the indeterminacy gets here a further characterization with respect to the natural language case: the reference within the Web is clear. The basic mechanism of URIs resolution will find for every label the exact resource to which it applies. The problem still holds, however, when one attempts to scale to external world, and tries to guess to which aspect, of the material entity corresponding to the web resource, that label would have been attached. The problem then lies now in the mis-correspondence, the imperfect correspondence between external, human, material reality and the objects that have their "existence" in the Web, that is resources (say pages, files), identified by their URI, and data as far as they are structured as resources, i.e. somehow wrapped and provided with a URI. For instance, the URL address of the personal Web page of the author of this work might be used in a trivial ontology to point to him – to the author – though it is apparent that such a trivial computational ontology could never exactly point to the human being who wrote these pages. The mismatch is here. And with it also a plethora of other ontological puzzles, well known to many ontologists, which find in computational ontologies a rich soil to proliferate. We may consider, just to give the flavour, what could happen if one considers that this author has two personal web pages, one at each University hosting his PhD course, and maybe also a private personal web page, or a public profile on a social network. What would imply to enter also these other references in the trivial ontology? Is the entity corresponding to the author to be considered as multiplied by four? This particular point, the peculiar relationship between external material world and the virtual dimension internal to the Web, is extremely delicate and would deserve a treatment by far more careful than this. Therefore we must content ourselves here to have introduced it and tributed to it its role in the translational problems of ontologies – here comprised the translation issues concerning the passage from an ontology to the other, but also, and most of all, concerning the translation from virtual, Web world and material reality.

Once it is clear the origin of the problem, however, it is due to tackle it in the proper way. But we do not propose to further investigate in the folds of the problem of the double level of reality. Rather we think that is possible, to some extent, to perform an analysis similar – though within the quite strict limits of Web reality – to the enquiry that the field linguist would perform among native speakers in order to explore the pragmatic component of the meaning of the unknown terms. This pragmatic component indeed is unveiled by the behaviour of speakers and is accessible by observing the concrete use of the language, the consequences that it produces on the (natural, social) environment.

In our setting consequences are at both the levels of reality. Consequences at the level of web reality are immediate and typically consist in the referencing of a web resource. It then needs just to observe (look up) what the terms in an ontology refer to in the Web, and record the referenced resources as sets of resources that are compatible to some extent and under some, no better defined, respect. It is not far from the job that a field linguist would do: annotating pairs of terms occurrences and candidate references, aiming to progressive disambiguation by comparison with subsequent occurrences of the same term, so as to finally unveil which one is the particular aspect of reality that the term is intended to signal or talk about. The idea then is to inductively and extensionally build the concept corresponding to the terms of an unknown ontology, maybe in a perennial work in progress subject to progressive adjustments and corrections – and basically, this is what our attempt to adopt Ludics on top of OCSs aims to. After all, well known thinkers (Quine himself, partly and Davidson for instance) have proposed positions not too far from this one concerning natural languages, even for the explanantion of reciprocal understanding between native speakers of a same language.

On the other hand, to track consequences at the other level of reality, scaling up to external human reality, requires human intelligence. Nevertheless one might still consider to pursue pragmatic aspects up to the boundaries between the two realities. For instance, based on the fact that web agents are to perform tasks on behalf of humans, a way to track consequences in the material reality would be to look for feedback consequences coming back to the Web reality. Typically this idea may take the form of a *learning agent* that receives some feedback from its master. But such a discourse shifts into engineering issues rather than philosophical. So let's take it apart for possible future applied research.

In any case, if there were not even a vague, loose correspondence between objects of web reality and entities of the material world, produced by more or less detailed descriptive ontologies that bridge the two levels of reality and thus let glimpse some approximatively one-to-one correspondence of objects, it would be absolutely pointless any effort to track and record term occurrences and their references. Likewise, the similar task would be pointless for the field linguist, if behind the terms of the new language there would be nothing like the concepts to grasp. Concepts indeed, independently of the actual existence of the conceptual entities that they call for, actually play a crucial role for communicating meaning, intentions. The concepts of computational ontologies then, that is the intensional counterparts of the extensional matter made of resources, should be considered as links, connections from the web reality to the concepts that appear in and form human knowledge. Indeed, and to conlude, computational ontologies for information systems are representations of knowledge, that is secondary representations of material reality, and consequently "secondary ontologies". Their final reference is to be found into human knowledge of the material world, not into material world itself. Curiously, this looks like the companion of the inversion of the roles between language and reality, pivoting on the notion of ontological commitment, that we have remarked above, while observing the purpose of assessing the ontological commitment of scientific theories on the one hand (as in most practice of analytic philosophy) and of Web ontologies on the other.

For the time being these peculiarities of computational ontologies for information systems seem to be sufficient reasons to accept the call to approach from a philosophical point of view the issues and problems of knowledge representation in the World Wide Web. These also form the basis for the final discussion, in next chapter, of the insights that our proposed approach, that of OCSs and Ludics, offers with respect to the issues and problems of knowledge exchange in the World Wide Web.

8. Ontology, Web and Interaction – Our conclusions

In this last chapter we aim to bring to some conclusion many issues that we have met and raised till now, often accompanied with a promise for a deeper and more careful treatment in this last, philosophical part. Now we are here and it is time to honour those promises. We organize this last chapter around three main themes: in the next section, devoted to the relationship between language and reality, we will attempt to accommodate and tidy up a number of questions that we have left open. In the second section we will attempt to conlude the compressed history of philosophical ontology by proposing an additional dimension for the comparison between ontological positions, that is the openness to other ontologies, the sensitivity to competition and / or collaboration between multiple, alternative ontologies. In the third section we will propose some concluding words about the theme of interaction, with particular consideration for communication as it appears as a fascinating form of interaction where almost every other issue that we have dealt with in the present work has a role to play. There is also one last, fourth section. There, we will briefly highlight a few aspects of this research that we consider most worthing of further development – as we consider them not only interesting, but also useful and maybe necessary for the complete unfolding of some ideas concentrated in this work.

This whole chapter however, in spite of its name, does not really propose conclusive, final positions. Rather, it proposes hints, suggestions that arise from a reading of Web-related issues that uses as interpretation keys, maybe a little daring, philosophical categories that we deem as illuminating. Thus, there is no aim to give solutions or to present the best way to approach any problem. We just suggest alternative points of view, contributions to a debate, that actually is not so lively concerning most of the philosophically important issues related to the Web, that surely would deserve futher elaboration before they could be presented as definite positions.

8.1. Semantics: meaning and interpretation

If we consider a basic notion of meaning according to which the meaning (of a statement, an expression) is that to which the understanding leads, then we may count, with respect to the arguments encountered till now during this work, three dimensions of meaning somehow accessible from within the World Wibe Web environment:

- a first dimension of meaning may be directly compared to the meaning of human languages. It deals with the intensions of concepts and relations between concepts, as indeed Semantic Web ontologies attempt to define them and to make them also somehow accessible to machines. This dimension could be *grosso modo* associated with the notion of *sense* (*sinn* in that tradition of German philosophy, and later on of analytic philosophy, that is rooted in Frege) especially as far as it is conceived as opposed to the other dimension known as

- *denotation*, which indeed we may consider as our second dimension of meaning. According to that tradition, a language is related to reality *via* a reference relation that associates terms (and larger constructs of the language) to objects in the real world[1]. On this end of the relation one finds the *denotatum*, that is the object corresponding to some term. Thus, we have roughly (re)covered the distinction between the intensional part of a knowledge base and its extensional part. But with respect to the Semantic Web we must note that denotation gets somehow splitted into two distinct realms that provide the range of the reference relation:
  - the external, material world that human beings perceive and know;
  - and the virtual world, inside the Web.
- and finally we consider the third dimension of meaning as logical interpretation, that is as the identification and understanding of a set of constraints to the possible models that a logic-aware agent (like a reasoner, i.e. an inferential engine, or a Semantic Web agent) would consider in order to evaluate an assertion (or a whole theory) – cf. Guarino's definition (p. 54) of ontology as expressing the ontological commitment of a language. To this dimension by the way pertains also our proposal concerning the Ontological Compatibility Spaces, that offers another way to give logical interpretation of knowledge base in the Web – and that allows to leave aside complex (and sometimes cumbersome) intensional interpretations.

Typically most people think of the semantics of Semantic Web as residing in the dimension of sense, i.e. of the meaning of natural languages, to be attained for instance when performing keyword based searches on a text based search-engine, so as to enhance both accuracy and recall thanks to a proper handling of and synonyms. This is not completely wrong – actually many, if not most of, efforts to achieve Semantic Web go that direction – but it is neither really right. Indeed, Semantic Web ontologies are designed to "teach" computers to properly handle human concepts inasmuch as these last are used to classify data and information in the Web, but not to give machines the capability to chitchat with us in natural language – though someone had this idea[2]. More seriously, the point is that ontologies are intended to enable a whole class of

[1] We are apparently simplifying this conception of meaning. For instance, the reference mechanism is not expected to work in this way for any kind of terms and expressions of a language. However we are not interested in giving more than a glance on this position about which everything has already been told, since on the contrary we aim to better illustrate an alternative position.

[2] We think of Doug Lenat and his *Cyc* project, dated 1984, for the production of a comprehensive ontology and related KB of common sense, sufficient according to him to make a computer to reason like a human being.

computer operated services – like the ones that one may figure out based on the interactive processes between Semantic Web agents that we have described in chapter 6 – for which the mastery of natural languages, or of intensional knowledge, is not really needed.

Moreover, if one tries to deal with the semantics of ontologies in the "conceptual" dimension (of intensions) then one eventually stumbles upon the traditional problems (and puzzles) of philosophical research on ontology, dealing with universals, properties and the like. We are not in the position to say whether this is a bad or good thing, we simply consider that even though, on the one hand, the experience with formalized ontologies for computer systems may have somehow stimulated ontological research on particular issues, on the other hand, however, the use and reliance of Semantic Web on such artefacts does not require to scale up to this dimension, which is rather a field of inquiry for speculative research. That which the ontologies must however provide is the approximation, as close as possible, of the meaning of human concepts by constraining the possible uses of the resources classified according to those concepts, so that the final, overall behaviour of some computer system could be deemed "intelligent", i.e. coherent with the nature of the resources dealt with. Hence, rather than semantics, as the name of Semantic Web somewhat unfortunately suggests, one should think of all this effort for enhancing the possibilities of exploitation of the information immersed in the Web as a matter of pragmatics, due to the effective use of information on the part of machines, or logic, due to the fundamental role of Logic to achieve such a result. At least, this is the conclusion that we support and to which we desire to lead the reader in these last pages. Indeed, to us, the really important dimensions as regards Semantic Web are those of reference (denotation) and most of all of (logical) interpretation. To this last, in particular, we hope to have somehow contributed with our proposal concerning the Ontological Compatibility Spaces – whereas we will focus in the third section of this chapter on a similar assessment of the value of our other proposal, concerning Ludics.

Before we shift to the level of logical interpretation, however, we still have some observations to propose about the couple sense and denotation, and the working of the reference relationship within the special environment of the World Wide Web. In particular we may note that in the Web there is a sort of collapse of sense and denotation as far as one is able to reach for the exact objects corresponding to the denoting terms occurring in a statement, be they concepts or individual resources – we may say now universals or substances respectively. In both cases indeed they can be deterministically reached by means of URIs (even better if they are URLs). Nice to observe, then, in the Web universals (reprsented as concepts) exist in the same way as objects do: they are all terms of the same nature, in an artificial language. If it were not too naive an observation, we could even say that this suggests a pretty realist (Platonist) approach to ontology for Web agents – roughly speaking, everything that has a name (an URI), and thus can be said, also exists in the Web. However, in the same line of naive considerations we will see also in which sense Semantic Web knowledge representations can be considered also from a constructivist point of view. Obviously the point of such considerations is just to use philosophical notions to quickly introduce ideas as just hints for further discussion. Thus, to consider the realism of Semantic

Web ontologies is a sort of provocation that aims to unveil the artificiality of their perfect semantics – since every term has its corresponding *denotatum* – that ensures a perfect treatment of every expected operation within the system that relies on a given ontology. "Expected operation" is to say every operation that has been foreseen at the time of designing the ontology and the applications using it. But, on the other hand, such perfect semantics is not able to cope with the unexpected, with anything really new. Such "realist" systems cannot but always repeat what they already know – that is what they have been explicitly programmed for. After all, the act of programming is the defintion of the semantics of an automated information system (like about any other computer system): on the programming indeed will depend the intelligent behaviour of the system[3].

Earlier in this work we have already spent some words about the novelty that the open environment of the Web has promoted in the field of Artificial Intelligence by presenting a scenery where there is not only one single intelligence (the artificial intelligent agent) but many agents that interact – in many different languages too – and we will come back on this in a while. But here we must consider the novelty that such setting brings to the notion of meaning and to the working of the reference relation. Frege (1848-1925) and many others after him have crystallized the notions of sense and denotation as two things pretty different, lying side by side. Along with this assumption, also the distinction between syntax and semantics in logic has became one of the traditional *tenets* of twentieth century, with the syntax playing the role of the sense (the process tha leads to the meaning) and semantics playing the role of denotation, signalling the corresponding objects (claimed to be) different from the objects manipulated by the machine during the process. The distinction syntax vs semantics has marked the development of computer science, being actually a fundamental element of its success: typically machines can effectively operate their functions by handling the syntax of some internal calculus and provide an external output that is meaningful thanks to a correspondence that is explicitly and *ad hoc* defined between syntax and semantics, for instance by assigning types to data – but such assignment of types is precisely the *deus ex machina* that puts the meaning in a "mechanic" system that does not produce it on its own. It is the usual technique: a machine acting as if it were intelligent, as if it really knew anything of the value that human users attach to the output that it produces. Actually, this point is a very delicate one from the philosophical point of view, since finally it leads to puzzles like the mental experiment of the chinese room, by Searle, and after all it deals with another huge question (what is intelligence) beside the one that we have started with, that is the relationship between language and reality, and how the notion of meaning fits in-between. We do not dare to tackle also this additional dimension in these few pages concluding a research work that focuses on something slightly different. We rather prefer here to consider at least one alternative conception of meaning that makes room for something else to account for meaning, something different from the reference relationship between language and reality. In fact, we too look for a broader, more complex and more in-

[3]By the way, it is interesting to consider how much the notion of *behaviour* proper to Ludics refers to a completely different conception of semantics, and more generally speaking of meaning in our World Wide Web setting.

structive conception of meaning, and we believe to have found a good track to follow by adopting Ludics to study communication between web agents. Ludics indeed claims openly for the abandon of the rigid distinction between syntax and semantics – which appears artificial (and artificious) in computer science, though very useful and effective – and leds us to consider a more general proposal for a logical understanding of the relationship between a language and the reality that it refers to according to which the meaning may emerge from the *use* of the language. It is precisely that which we have signalled in chapter 6, when talking about the emergence of concepts from interactive query-answer processes. However, it is interesting, and somehow due, to consider also alternative accounts of the meaning that have at first sight nothing to share with this proposal but that, anyway, have considered the use of language for communication as the key element for studying its meaning.

A name to mention is then surely Wittgenstein (1889-1951). His thought is traditionally presented as neatly divided in two periods, the "earlier Wittgenstein" (concentrated in the only book he published, the *Tractatus Logico-Philosophicus*, 1921) which was, then, considered next to the positions of the Circle of Vienna – the *milieu* from which most of logical positivism came out – and the "later Wittgenstein" that is presented as a thorough rethink of his philosophy. Accepting this rough division, we may say that, as regards the themes of this section, the earlier Wittgenstein contributes to mark the way for the fomalization of the language that will become one of the leading ideas of logical positivism. Thus, he moves comfortably in the line of those who investigated meaning by looking for the denotations of terms and more complex expressions. And he also was taken as an example figure by neo-positivists claiming for the cleansing of the language to be used in science and philosophy from everything was not clearly referencing something actually existing in the world (as we said of neopositivism in previous chapter). Merged with the effort for perfect logical formalization of language, this is the obsession for the perfect language typical of neopositivism, and we know how that project has wrecked. But we can nevertheless find traces of those convictions in some aspects of analytic philosophy: consider for instance Quine and his effort to found ontology on the criterion of ontological commitment of the language (of sciences). In Quine but also in neopositivism and any similar philosophical position, the meaning is reduced to the reference relation, and the (interesting part of) language is just that which neatly gives access to quite well identifiable objects in reality. By the way, we could see here also, finally and ironically, a sort of success of these positions precisely in Knowledge Representation ontologies, where reality (albeit the fictitious reality of a software system, or maybe a classic AI system) is in a 1-to-1 correspondence with the logical language that describes it, since after all this language also defines and builds that reality. Nevertheless, we are by far more interested here in considering what the later Wittgenstein said about language and meaning: that is the time of the *language-games*. With language-games Wittgenstein supports the idea that the central issue with the language is its use, and simply discards the referential approach. The notion of language use in Wittgenstein is actually quite a different position from that of interaction in Ludics – there are maybe more different aspects than commonalities, but it is worth to note that, beyond the notion of use, Wittgenstein too conceived the linguistic competence of a speaker as the sum of many different language-games with which he is acquainted, each one relevant to a different context wherein he acts (for instance, a language for family relations, a language for cultural frequentations, a language for work, and so on). This has interesting points of contact with the idea of the web of ontologies and of the learning activity of a Semantic Web agent that learns by exploring other agents' knowledge. Moreover, we would suggest to consider, behind any particular language-game also a corresponding ontology that participates in defining the whole store of knowledge of an agent that is able to play that game, i.e. speaks that particular language, i.e. deals with that reality: it brings along a KB compatible with that ontology.

To be honest however, many aspects of games *à la* Wittgenstein indeed do not hold with respect to Web agents' communication. In particular, whereas Wittgenstein tended to highlight the mechanisms that subsequent work has made known by the names of *speech acts*, on the contrary we have to focus on the basic denotative function of the language – since Web languages are designed precisely to manipulate symbolic objects clearly identifiable, and not in order to *directly* perform any other kind of action. Nevertheless, through the manipulation of symbolic objects also web agents could be able to produce actions or trigger events that eventually have some consequence also in the external material reality (think of hotel booking, car renting and the like), as speech acts do. As a consequence, then – besides the approximations that we can reach for thanks to OCSs cliques (that collect individuals sharing some sort of common type) and to corresponding Ludics behaviours – it seems possible to consider yet another way to investigate the meaning of terms used as folksonomy tags and concept-names in ontologies by observing what they may trigger with respect to the external, material world, though it was not the intent of such languages to really "come back" from the Web virtual world to the external, material reality. We admit that it is quite an imaginative suggestion. In fact, to consider speech acts performed *via* web agents could be just a special case of human beings' speech acts, not deserving any special treatment since after all the intentional content of the act (an interest to satisfy) belongs to the human master of a web agent, not to the agent itself. Anyway there is an interesting point that emerges from such an observation. The set of resources that can be retrieved by an agent after a data retrieval query over the KB of some other agent is a collection of symbolic objects that have some correspondence with objects (in a very broad sense) in the external world. Actually, to say precisely what is this correspondence with the external world is the most delicate issue concerning the Web. From this, indeed, a number of problems arise, typically related to the problem of identity, multiplicity of references, co-reference and indeterminacy of the *denotatum*, think for instance of the quick example that we used in the previous chapter about multiple personal webpages of this author. We cannot adequately tackle these problems in this work since we focus on a pretty different issue (the possibilities of communication, of successful interaction between web agents) that does not require necessarily to ever get at the external world. We rather prefer to signal the reader some much more serious presentations of the issues concerning the correspondence between Web objects and external world, like [Miklós et al., 2010] and [Bouquet et al., 2009]. Nevertheless, the point of the observation above, that we propose here just passing on, is that through the lens of Ludics we could get a new insight also on this subject.

Indeed, the symbolic objects of Web reality (be they URI-strings or other datatype strings)[4] that are retrieved after a particular querying session between web agents form some sort of concept[5] (in the sense of OCSs cliques or Ludics behaviours) only because each of them is a "legal", logically correct answer to a same question (the query). The idea then is just to consider, also for a better understanding of the dignity of Web reality, an alternative point of view: besides a conception of meaning as an exact function that should lead from terms and expressions of the special Web languages to objects existing (in some way) in external, material reality, we could consider also a more flexible conception of meaning as something not definitely fixed, that results as a (possibly temporary) agreement between two agents that communicate and are committed not directly to the existence of something, but to the achievement of their tasks, and the language(s) that they use is just an instrument to this aim.

Apart from these tortuous "constructivist" considerations about the discovery of meaning in the Web and from the Web, we should finally account for a very neat realist position that we deem really penentrating for a proper setting of the reference relationship between language and reality – actually, this realist position will allow us to better explain the significance of the constructivist way of discoverying meaning in the Web, since the founding ideas are not at all contrasting, rather they integrate quite well (see below and cf. section 3 of this chapter). The realist position we mean is the one exposed by Smith, for instance in [Smith, 2004]. Smith attacks the notion of *concept* as the central notion for Knowledge Representation and proposes on the contrary to take back, at the core of every effort in ontology – that is in particular also in ontologies for automated information systems –, the notion of *universal*. Concepts are depicted as an unnecessary intermediate level in knowledge representations: if ontologies are based on concepts, which are ideal products of human mind, then such ontologies are second level representations (representations of representations), too far from reality. And indeed, we must admit, the definition by Gruber of formal ontology as the specification of a shared conceptualization tends to such second level (and maybe second hand) representations. The point of Smith is that there is no reason why the objects of formal(ized) ontologies should be mental representations rather than the *types*, the *properties*, the *universals* that ontology has always dealt with. This is even more apparent if one considers Description Logics ontologies. The term "concept" in DL ontologies indeed is short for concept-name, and to every concept-name is associated in principle some concept-description. The significance of a concept in DL ontologies indeed is its logical definition (the concept-description with which it is provided), that which makes of the concept(-name) a logical type that may be assigned to resources (individuals) so that the manipulation of resources on the part of special programs can be decorated with semantics – and machines may operate tasks

[4]To limit the complexity of this discourse, that we could not entirely face here, we leave aside the case of resources whose existence is entirely within the virtual reality of data and data streams (videos, music tracks, computer files in general) and do not consider the "nature" of the possible corresponding objects, as indeed the point of the discourse is precisely to step aside these questions so as to observe other aspects that usually are not considered.

[5]We recall that such concepts are only *locally* meaningful, that is their significance is scoped to the KBs (and corresponding ontologies) of the Web agents participating in the querying session.

that seems to be achieved by intelligent agents aware of the nature (their ontological type) of the resources. We agree with Smith on this point – that the notion of concept in ontologies is misplaced – and indeed, albeit we have used (and may continue to use) the term "concept" according to the tradition of KR, we have used it as an alternative to type, or even *tag*. It is significant to this respect that we have put on the same level, and dealt with in the same way, both concept-names from ontologies and tags from folksonomies: to us they are all markers for types to be assigned to resources in the Web. Now, the point of this parenthesis on concepts and universals, and the claim for the realist position of Smith is finally that he attempts to properly readdress the reference relationship from the language to the objects existing in reality – avoiding intermediary steps such as mental constructs known as concepts.

8.1.1. Logical interpretation

Smith, however, provides us with some interesting arguments also as regards the other dimensions of meaning that we have considered as relevant to the Web (and Semantic Web obviously) at the beginning of this section. If the *denotational* dimension of the meaning of Semantic Web knowledge represetation languages gets deterministically resolved in the domain of Uniform Resources Identifiers (what we may call the realism of Semantic Web ontologies), then the dimension of *sense* of such languages could be fully resolved in their *logical interpretation*. Let us recover directly from Smith a quite clear account of what is this dimension of meaning as logical interpretation:

> Each DL concept description represents, with respect to any given interpretation, a collection of objects that are postulated as sharing the property that is specified by the description. Even this does not provide an anchor for concepts in external reality, however, for the objects in question may be (and standardly are) merely abstract mathematical postulates. Thus when it is said that DL provides the terms we use in ontologies with a 'precise semantics', then we should bear in mind that the sense of 'semantics' at issue here involves recourse to a mathematical abstraction that is far removed from our normal understanding of semantics as relating to the interplay between terms, meanings and corresponding entities in reality. [Smith, 2004]

First of all a note to the quoted text: where Smith says "any given interpretation" he means any assignment of values to constants of the language (predicate and individual symbols, as we observed in chapter 5, are indeed logical variables, known as constants at First Order). That the logical interpretation of ontologies is something like that, clearly, is not just Smith's or our opinion: it is precisely thanks to such interpretation that automatic reasoners may derive theorems from ontologies (and KBs too). Thus, we can see that along with a problematic reference relationship that should associate concept-names with objects in external reality – corresponding however to quite a natural notion of meaning – ontologies are given a (seemingly) less problematic interpretation in logical structures, which, on the other hand, are pretty not natural. The unnatural aspect does not depend on the fact that, after all, it is "syntactic semantics", or semantics syntactically generated – also Smith, by the way, speaks of concepts (in ontologies) as syntactic entities (*via* the understanding of concept as a DL concept description). It depends on the features of the logico-mathematical objects that the interpretation is built on. Basically they are sets (of Set Theory). We have already spent, in the first part of this work, a few words on sets; now we are going to make it clear why they are not the best choice for representing concepts or, even worse, ontological universals. The problem with sets is that sets are perfect, and indeed fit well for mathematical objects – which are perfect too: perfectly defined, they exist *by* their definition, unlike any other object that is not produced by a science but exists on its own. With more details:

- sets are "full", that is a set automatically contains all the elements that have the property that defines the set (its characteristic);
- sets are fixed, that is a set cannot suffer any change in its definition;
- sets are invariable, that is a set does not suffer "turnover" of its elements.

All these aspects clash with the reality of material world, but also with the reality of Web resources and automated information systems. For instance, the set of employees counted in the KB of a large company is everything but a set: the definition of the set may be subject to modifications (e.g. in case of a change in the company denomination the set of employees could change from Employees of ACME Ltd. to Employees of ACME Corp. – apart from the triviality of the example, it is a change in the characteristic, the defining property of the set); the current collection of individuals appearing as elements of that set could not always correspond to actual reality (e.g. in case of a newly hired employee, there may be a delay before the KB is updated); and the possibility itself of hiring new employees as well as the other possibility for employees to change their job or retire, signals that the set of employees must suffer quite an important turnover of its elements. After all, people and companies live in a world subject to time (as also computer systems do), whereas sets stay in a timeless world.

If we look at the issue with a little more attention, we easily would say that in fact a property like "being an employee of ACME Ltd." seems not to be a good candidate for stating the existence of a corresponding set (in the strict sense of mathematics and logic for this term), in the same way as it does not look a good candidate to appear in a scientific theory – for which sets usually fit quite well. Obviously, the payoff for accepting such distortions, like having a set and a formal predicate out of that property, is the possibility to have a reasoner drawing inferences about it. So, the problem, really, is primarily with the transformation of properties of real individuals (universals, ontological types) into logical predicates, which a reasoner can handle, and, just as a consequence with their interpretation as sets since FOL finds in Model Theory its interpretation and this last builds largely on sets. Another "bizarre" consequence of an interpretation that passes through the reduction of a universal to a logical predicate is that as a predicate it can be combined, through logical operators, with any other predicate, thus generating more complex predicates for which, however, there is no guarantee that they stay meaningful with respect to external reality. For instance, it is licit to consider the intersection of the sets of employees and of machineries of a company as far as they correspond to predicates appearing in the logical theory to which an ontology (and relevant KB) describing the ACME Ltd. company is reduced, even though it has no point nor meaning with respect to reality.

By comparison with the shortcomings of this logical interpretation usually attributed to ontologies (especially DL ontologies), we may illustrate the answers and possible solutions that the alternative logical interpretation that we have proposed in the previous part of this work may provide. In particular, with the Ontological Compatibility Spaces,

- sets are traded for mathematical objects (the OCSs) with less "demanding" characteristics – no pretension for fullness and fixity, since OCSs are imperfect objects missing any intensional definition, they are only locally closed;
- OCSs do not pose constraints to the variation ("turnover") of their elements;
- the compatibility relation gives them a richer internal structure than (bare) sets;
- there is no place for predicates – compatibility itself replaces them all within OCSs.

As we have seen in chapter 5, concepts (or universal we would rather say now) appear as cliques, which indeed are the best approximation of concepts / universals based on the knowledge available in each KB. Maximal cliques (in all three variants that we have considered for the notion of maximal clique) approximate concepts / universals by collecting all the resources that are known to have that given characteristic / to instantiate that universal, within a given KB. The local dimension – opposite to the absoluteness of sets – depends precisely on the scoping of these approximations on the particular KB on which they are evaluated. Moreover, OCSs fit well a setting where many alternative ontologies are available – as in Semantic Web – and resources described in a KB with an ontology can be compared with resources described in another KB with a different ontology, still allowing for mixing of cliques, corresponding to the possible discovery of some common concept between the two ontologies. Indeed, once two (or more) ontologies are to be put together (merged, mapped, aligned . . . according to one of the "operations" on ontologies discussed in chapter 4), then also the local dimension of maximal cliques may grow, according to the rules of operations between Coherence spaces, so as to reflect the possible increase in knowledge about a concept / universal. It is the same principle that we find in action also in the case of Ludic-querying, where an agent builds up its knowledge of a concept / universal based on the answers that it receives from other agents, i.e. confrontating its own KB with those of the other agents and possibly enlarging the cliques that approximate some concept / universal whenever matching terms or compatible logical definitions are found, so that new resources are recognized as compatible with other already known resources.

8.2. The time for ontologies

We have just recalled here above the possibility to deal with multiple ontologies. Indeed, the compresence of many different ontologies, often partially overlapping as regards the domains that they are to describe, is a characteristic that Semantic Web inherits from World Wide Web and its "freedom of expression". Such a freedom on the one hand allows everybody – every organization, company, ... – to design, adopt and use whichever ontology they prefer in order to constitute the formal vocabulary to use for tagging data and share them over the Web; but, on the other hand, to this freedom is tied also the difficulty to have a working Semantic Web, for the simple fact that to have different systems to communicate it is expected to have first of all a neat reciprocal translation of their languages. To produce this translation is not a trivial task and it does not depend only on an adequate logical interpretation of vocabularies that are not given a clean FOL dressing like DL ontologies (think for instance of RDF schemata). Rather, it is a matter of "reverse ontology" (joking on reverse engineering) that requires a delicate process to unveil the ontological assumptions codified in a knowledge representation (or even a KB). This in its turn implies the recognition of universals (and possibly also of the particulars, the individual objects) existing in reality that the terms in the ontologies stand for. One can easily see here a twisting process from representation to reality and back to representation to provide mappings or similar arrangements that allow different systems to profitably exchange information. The bottleneck is clearly the passage to external reality, which requires inevitably the intervention of human experts at some stage of the process that attempts to associate real objects to the terms of a formalized ontology.

Before we get finally, in next section, to "praise" the benefits of pursuing a different strategy to account for communication on the Web, we can nevertheless linger for a while on the issue of multiplicity of ontologies as a value, not as an hindrance to a neatly working Semantic Web. The openness that makes room for multiple ontologies, indeed, is the most important novelty of Semantic Web (as a matter of formal representation of knowledge) with respect to 1970s and 1980s expert systems, which too were endowed with pretty rich knowledge bases, ancestors of today's ontologies, but which were accessed and exploited, as in any traditional computer program, by a single agent within a closed environment. And openness causes that many different, alternative and perhaps even "competing" ontologies are now available, and actually used, in the Web, with no viable way to precisely map each into the other ones. In our opinion, this is the reason for the implausibility (if not impossibility) of a success on the field of the Web for the traditional approach that relies on external semantics, given side by side to syntax – and indeed syntax it too. For it would require, as it does indeed e.g. in the case of Semantic Services and other web-based services, to provide given-in-advance mapping instructions between the ontologies of any two systems that should cooperate to compose a complex service.

From a more philosophical point of view, however, the issue of the multiplicity of ontologies is much more than this. It indeed provides us with another category to contrast ontologies (as philosophical, metaphysical systems) with each other and possibly attempt comparisons and classifications – which however we do not aim to produce here: we aim to signal this possible category, not to operate the classification of philosophical systems and positions with respect to it, for apparent reasons of convenience to this work. In the Web in fact we are automatically presented with an environment where multiple ontologies are available. Most of them are intended to cover just minimal portions of knowledge (domain specific ontologies), whereas a few others aim to become pillars, if not the core of Semantic Web by dealing with foundamental ontological (and philosophical) issues that typically are disregarded in the other ontologies. These last, “ambitious” ontologies are the foundational (or upper-level) ontologies. The objective that such ontologies aim to is to provide the “ontological” glue that would allow, in principle, all ontologies in the Web to be connected meaningfully (at least partially) with each other, thus composing a real web of ontologies.

On the contrary, in the history of philosophical ontology typically there is no room for multiple ontologies – at least up to pretty recent times. As a consequence, it can be taken as a relevant criterion to assess an ontology whether it may accept, or not, other ontologies to describe the same or another close domain, the same or another close portion of knowledge about the world. In past times, from Aristotle to recent times, each philosopher involved with ontological research had typically to define his own ontology. Ontologies by other philosphers were either a point of departure for defining something richer and more and better articulated, or competitor ontologies, so that the relationship among them had to be either deepening, elaboration, enhancement or competition, more or less open, in order to prevail and be accepted as the “right” ontology, maybe the one best complying with the empirical evidence. Present days on the contrary present us with some cases of *collaborating* ontologies, that is ontologies that deal not with the whole world, but just with portions of it, complementary to each other so as to produce a full, complete ontology only once put altogether. It is apparent in the case of many Semantic Web ontologies, like in the setting of a foundational ontologies and many limited domain ontologies, possibly modularly designed so as to allow for reuse of even small “piece” of ontology. Recovering from the previous chapter the notion of mesoscopical ontologies, one can easily note that the best candidates to form a network of ontologies are precisely the mesoscopical ones: they are purposely designed, even as single ontologies, in such a way as to allow for different accounts of a same reality, depending on the level of grain that one adopts to get things explained. Precisely because of the possibility to propose alternative accounts and explanations that are not necesarily in contradiction with each other, but on the contrary allow for a deeper overall understanding and knowledge of reality, the collaborative dimension becomes a significant aspect also for contemporary ontological research, and we mean here Ontology with capitalized “O”, i.e. the philosophical inquiry on what really exists. The scenery of “collaborating ontologies”, that is ontologies quite flexible and liable to integration with other ontologies, seems to us an interesting alternative to reductionist programmes. Whereas these last look for a single, ultimate ontology capable to explain and account for everything by grounding all the answers onto only one fundamental level of scientific observation, collaborative systems of ontologies could combine the universality of the eventually resulting complex system of ontologies, with the ability of any single specialized ontology to account for specific issues in the most convenient, natural way.

8.3. A matter of communication

In this section we aim to put together the variety of hints and suggestions sparse in this work which bring arguments to a discourse about communication as the central focus of our technical proposals and philosophical position. Indeed, with respect to the title of this chapter (Ontology, Web and Interaction), we have that the Interaction that we consider is the peculiar form of communication between automated information systems in the Web, and Ontologies are needed and studied as the "technology" that assists this communication. Some form of ontology indeed appears as a prerequisite of communication, as any single ontology (in the sense of Semantic Web ontologies, of course, but not limited to this) *determines* a language. We had noted in the previous chapter the inversion that occurs concerning the notion of ontological commitment with the passage from ontological commitment as a criterion for philosophical analysis (as in the analytic tradition) to ontological commitment as the property of a theory that constrains possible (correct) interpretations of a language. On the one hand, one might see in this a diminution of the importance of a philosophical formal instrument such as the criterion of ontological commitment, whose destination of use is turned, from inquiry on what actually exists, to binding formal languages for machines. But on the other hand we have that such a role of ontology allows to take a new look on reality by removing the conventional covering (like social and cultural conventions for instance) that hides the mechanisms of communication. We believe that this alternative perspective on the relation between ontology and language, proposed by the practice of Knowledge Representation, may provide contribute to attain a deeper understanding of the relationship between language and reality already in the dynamics of its own determination. In other words, it allows to point out to which objects, to which reality the communication commits. This holds clearly for Semantic Web agent's interaction, as we have already stated in chapter 6, where we have seen that whereas ontologies build the reality of a system (or an agent), the communication between agents produces the "conquest", the appropriation of the knowledge of the other agents, which is (the knowledge itself) the reality in which agents are to operate. That is, the exchange of information between agents (by means of the ideal interactive protocol described in chapter 6) leads agents to commit to new elements, objects and properties of the reality wherein they operate. But it would be worth considering analogous processes with respect to the general case of human natural language, and therefore with respect to material reality. It could give precious suggestions concerning how human beings too determine to which elements of reality to commit. After all it seems natural that it should be there too into play the principle that, like successful communication identifies correct use of the language, so the use of the language determines meanings that the behaviour of speakers (be they software agents or human beings) must abide by. In proposing such a reading of communication we may rest on the position of Lecomte and Quatrini (in [Lecomte and Quatrini, 2009]), who express the baseline of their attempt to interpret the semantics of human interaction in dialogues by means of Ludics with these words:

> We shall follow the same idea [as Ludics] by proposing that sentence meanings are given by the counter-meanings they are opposed to in a dialectical interaction. [Lecomte and Quatrini, 2009]

This is not to say that reality itself is determined by the use of language, and in particular by the successful interactive performances between agents or speakers. But the meanings that speakers share are "handles" that give access to objects in reality (through the mechanisms of reference that we have discussed above) that could be disclosed precisely during the communicative interaction.

We follow about the same idea as [Lecomte and Quatrini, 2009] by proposing, with respect to Semantic Web agents' interaction, that meanings (of folksonomy tags and ontology concept-names) are given by the answers they are opposed to in a dialectical interaction, provided that these meanings are requested through *queries*. These last can be answered, in our setting, in two slightly different ways: by showing the logical definition of a concept-name (where available), thus determining the sense; or by pointing out the resources (the individuals in an A-box or the resources registered and described within a folksonomy) to which the tag / concept applies, thus determining the referenced objects.

Perhaps we might have arrived to the same position also from a different way: looking at that which happens "naturally" with folksonomies, i.e. during the normal use of folksonomies on the part of human users – who are members of a community that catalogues resources sticking tags on them – and abstracting from that. In chapter 3, indeed, we have illustrated in which sense tags, and also the labels they carry, can be considered as just marks of use. The next step is to consider them as handles to grasp objects in some reality: it can be either the Web reality of webpages, digital multimedia files and the like, or the reality of external world – of flights and hotels to book for instance. Obviously, whereas the reference mechanism works perfectly (apart from 404 errors) with respect to Web reality, it is highly unpredictable with respect to external reality. The final step is then to consider whether, and to which extent, the experiences of successful interactions with a folksonomy (query-answer exchanges between web agents as well as human browsing by tags) may lead to the definition of a (perceived) world, a universe of reference that is scoped within the corresponding community and tagging environment.

8.4. Next steps

We consider this whole work as a work in progress. We have approached it with the conviction to be in a favourable position to attempt a cross-fertilization between quite different research areas: Semantic Web, Linear Logic, software agents, philosophical ontology, ... all put together under the umbrella of *communication*, meant as interaction aiming to share information about what is there in some more or less constrained universe of reference.

We cannot say whether this cross-fertilization has achieved any success now. We rather see a number of themes and delicate issues that have been disclosed, or have been approached from an alternative point of view, and that now would deserve further study. We only highlight here the main directions that we would like our future work to follow:

- on the logical and technical plan, work is needed to reach for a clean integration of the interpretation of ontologies and folksonomies as Ontological Compatibility Spaces with the modelling of Semantic Web agents' interaction by means of Ludics;
- secondly, the idea of a lightweight query protocol, capable to support interaction between Semantic Web agents even when they are not purposely designed (basically by making them to speak the very same language) to exchange information would deserve a better, more precise definition of what this protocol should actually be;
- on the philosophical plan, then, there is a number of open issues. To be short, all that we have proposed in this last chapter would deserve both deeper and broader consideration.

For breadth we mean both comparison with other positions and points of view, and also consideration of other aspects that we have not taken into account here. In defence of the positions expressed in this work, however, we recall that the original intention of our philosophical considerations is to offer stimuli and suggestions to accompany and feed an interest in the Semantic Web as a matter that is not a privilege of engineers and computer scientists but that, rather, would deserve large contribution by philosophers and humanists – besides, obviously, the motivation of attempting to provide a comprehensive reading of the technical proposals that we have presented in this work. Among the philosophically most fascinating issues that we would furtherly consider we count in particular

- the issue of communication between agents in the Semantic Web – we would like to better describe the learning process that we have sketched (in chapter 6) for Semantic Web agents, and to properly assess to which extent this can be legitimately considered as a way to discover reality;
- and the fundamental relationship between language and reality, focusing on the role of ontology in-between – looking for a sort of equilibrium point where ontology ceases to be an artefact that supports a language and faces directly reality, thus distinguishing its role as an explantion of reality from its role as a definitorial theory.

Bibliography

Aristotle. *"De Interpretatione"*. University of Virginia Library Electronic Text Center, 1928. Translated by E. M. Edghill.

Aristotle. *The Categories*. Project Gutenberg, 2000. Translated by E. M. Edghill.

Paolo Bouquet, Themis Palpanas, Heiko Stoermer, and Massimiliano Vignolo. A Conceptual Model for a Web-Scale Entity Name System. In *ASWC*, pages 46–60, 2009.

Immanuel Kant. *Lectures on Metaphysics*. Cambridge University Press, 1997. Translated by Ameriks Karl and Naragon Steve.

Alain Lecomte and Myriam Quatrini. Ludics and Its Applications to Natural Language Semantics. In *Logic, Language, Information and Computation. 16th International Workshop, WoLLIC 2009. Proceedings*, pages 242–255. Springer Berlin Heidelberg, 2009.

Zoltán Miklós, Nicolas Bonvin, Paolo Bouquet, Michele Catasta, Daniele Cordioli, Peter Fankhauser, Julien Gaugaz, Ekaterini Ioannou, Hristo Koshutanski, and Antonio Maña. From Web Data to Entities and Back. In *CAiSE*, pages 302–316, 2010.

Kevin Mulligan. Métaphysique et ontologie. In Engel P., editor, *Précis de Philosophie analytique*, pages 5–33. Presses Universitaires de France, 2000.

Frédéric Nef. *Logique, langage et réalité*. Editions Universitaires, 1991.

Frédéric Nef. *Traité d'ontologie pour les non-philosophes (et les philosophes)*. Folio essais, 2009.

C. K. Ogden and I. A. Richards. *The Meaning of Meaning*. New York, Harcourt, Brace & World, Inc., 8th ed. edition, 1923.

Roberto Poli. *Ontologia formale*. Marietti, Genova, 1992.

Roberto Poli. Descriptive, formal and formalized ontologies. In Denis Fisette, editor, *Husserl's Logical Investigations Reconsidered*, pages 183–210. Kluwer Academic Publisher, 2003.

Willard Van Ormond Quine. On What There Is. *"Review of Metaphysics"*, 2:21–38, 1948.

Barry Smith. Formal ontology, common sense and cognitive science. *International Journal of Human-Computer Studies*, 43:641–667, 1995.

Barry Smith. Truthmaker Realism. *Australasian Journal of Philosophy*, 77(3):274–291, 1999.

Barry Smith. Ontology. In Floridi L., editor, *Blackwell Guide to the Philosophy of Computing and Information*, pages 155–166. Blackwell, 2003. Oxford.

Barry Smith. Beyond Concepts: Ontology as Reality Representation. In Achille Varzi and Laure Vieu, editors, *Formal Ontology and Information Systems. Proceedings of the Third International Conference (FOIS 2004)*, pages 73–84. IOS Press, Torino, Italy, 2004.

Barry Smith. Ontologia e sistemi informativi. In *Networks*, volume 6, pages 137–164. Servizio Web Italiano per la Filosofia, 2006.

Barry Smith and Christopher Welty. FOIS introduction: Ontology – towards a new synthesis. In *Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01, pages .3–.9, New York, NY, USA, 2001. ACM.

Paul Studtmann. Aristotle's Categories. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. 2008. `http://plato.stanford.edu/archives/fall2008/entries/aristotle-categories`.

A. Elements of Linear Logic and Ludics

A.1. Coherence spaces

Definition 11 (Coherence spaces) *A coherence space X is defined by its:*

support *: the underlying set of points, noted $|X|$*

coherence *: a binary, reflexive and symmetric relation between points of $|X|$, noted $x \frown_X y$*

A subset a of $|X|$ whose points are all pairwise coherent is called clique*, and is noted $a \sqsubset X$.*

Equivalently, an alternative definition is based on the notion of clique:

Definition 12 *A coherence space is a set (of sets) X that satisfies the two conditions of* down-closure

- *if $a \sqsubset X$ and $a' \subset a$, then $a' \sqsubset X$*

and binary completeness

- *if $Y \subset X$ and $\forall a_1, a_2 \in Y\ (a_1 \cup a_2 \sqsubset X)$ then $\bigcup Y \sqsubset X$.*

A.2. Linear Logic sequent calculus

The sequent calculus **LL** is the system of deductive rules of Linear Logic. For A, B formulas and Γ, Δ finite multisets of formulas, i.e. finite sets of occurrences of formulas, the rules of **LL** can be divided in two groups: identity and logic (without structural rules).

Identity / Negation

$$\frac{}{\vdash A^\perp, A}\ (identity) \qquad\qquad \frac{\vdash \Gamma, A \quad \vdash A^\perp, \Delta}{\vdash \Gamma, \Delta}\ (cut)$$

Logic

$$\frac{}{\vdash 1}\ (one) \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \bot}\ (false)$$

$$\frac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}\ (times) \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\ (par)$$

$$(no\ rule\ for\ zero) \qquad \frac{}{\vdash \Gamma, \top}\ (true)$$

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B}\ (left\ plus)$$

$$\frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \,\&\, B}\ (with)$$

$$\frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B}\ (right\ plus)$$

$$\frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A}\ (of\ course) \qquad \frac{\vdash \Gamma}{\vdash \Gamma, ?A}\ (weakening)$$

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}\ (dereliction) \qquad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}\ (contraction)$$

$$\frac{\vdash \Gamma, A[t/x]}{\vdash \Gamma, \exists x A}\ (exists) \qquad \frac{\vdash \Gamma, A}{\vdash \forall x A}\ (for\ all)$$

A.3. Ludics

A *design* is an abstraction of a focalized proof in sequent calculus. In order to see more precisely what is the alternance between positive and negative actions in such a proof, we should look for a moment into a sequent and a sequent calculus proof. A sequent is a collection of logical formulas, that are progressively analysed and decomposed (simplified) into their subformulas in a process that is typically represented as a tree, with the root at the bottom which is the formula to be proved, also called *base*, and at the end of decomposing branches, as leaves, some axioms may be given. Sequent calculus, thus, provides the proof of a formula be reducing it to the truth of its premises and showing how truth is preserved through the series of logical operations that brings up to the final formula. A focalized proof somewhat compresses the resulting tree by grouping all actions (that is application of a logical rule of sequent calculus) of a same polarity that happens before an action of the opposite polarity. Every action is actually a decomposition step of the formula that is chosen as the *focus* of the action for that "round". Since the main connective of a negative formula is a negative one, the "policy" with polarized proofs requires the formula that contains it to be decomposed immediately – it is the most convenient choice due to reversibility of the action. All subsequent negative decomposition steps are counted within the same step, so that next step will require a positive action. In case of a positive action, on the contrary, the focus may be chosen among the formulas produced by the previous action. Briefly a decomposition step – corresponding to the application of a series of logical rules of same polarity in a sequent calculus – consists in

- either choosing a positive formula and decomposing it into a set of negative subformulas, thus producing a set of sequents (one for each negative formula),
- or decomposing the negative formula into a set of sets of (positive) subformulas, thus producing a set of sequents (one for each set of positive subformulas).

Now, to come back to Ludics, we have to substitute formulas with their addresses (or *loci*). As a consequence the decomposition path of a formula by its subformulas gets replaced by the "subaddressing" of an address, that is the progressive addition of a suffix (called *bias*) to every address that is "accessed" at each action.

Every *action* in Ludics then is an abstraction of some decomposition step. It has a polarity (positive or negative) and is noted by one focus, i.e. the address on which the action is focused, together with a finite set of biases, called a *ramification.* A special (only positive) action is the *daimon*, noted ✠. The base of a design is a set of addresses noted $\Upsilon \vdash \Lambda$, where Υ is either empty (and the base is positive) or a unique address (and the base is negative) and Λ is a finite set of addresses. A *chronicle* is a sequence of actions with distinct focuses that can be read on a design. In a chronicle positive and negative actions alternate, and the first action depends on the polarity of the base:

- if the base is positive, the first action is positive too and it can be either a normal positive action (on a focus plus some biases) or the daimon;
- if the base is negative, the first action will be negative too.

In particular, if a daimon appears in a chronicle, it concludes it. The focus of every action (which is not a daimon) is an address produced by one of the previous actions or it is present in the base of the design. The focus of a negative action is either in the base (and then the action is the first in the sequence) or it is produced by the action just before in the sequence.

A design on a basis $\Upsilon \vdash \Gamma$ then is a set of chronicles such that

- the set is prefix closed (chronicles start at the base and develop – even infinitely – by adding suffixes to the addresses);
- the nodes where the design branches are positive actions;
- actions occurring just after a branching with distinct foci for the different branches, must have distinct foci for all the rest of the sequence of actions (this allows to split chronicles);

- the leaves of the tree thus generated are positive actions;
- whenever the base is positive, the set of chronicles on it cannot be empty.

There are two main ways to represent designs: the first one recalls quite closely the trees of sequent calculus, focusing on the decomposition steps (the subaddressing of loci); the second one takes better care of the intuition of designs as sets of chronicles and draws a design as an arborescence, focusing on the actions (and somehow hiding the heavy accumulation of suffixes). For our examples we will prefer this last.

Cut-reduction (cut elimination, or normalization) – that which we have promised to reduce Semantic Web agents' interaction to – is the process occurring within special nets of designs known as *cut-nets*. A cut-net is made of a set of designs satisfying the following conditions concerning their bases:

- the loci in the bases must be pairwise disjoint or equal;
- loci which are equal must occur once in the left part of a base and once in the right part of another base. Thus they form a cut;
- the net resulting from cutting the bases must be acyclic and connected;
- one main design can be isolated, whose base either has the left part that does not get cut with any other base, or it has no left part.

We can now sketch how cut reduction proceeds between designs in a cut-net. Let $\mathfrak{D}$ be the main design in a cut-net, with first action (σ, I) or daimon:

- if the first action is the daimon, then the result of cut reduction is the design reduced to the daimon;
- if the first action is (σ, I) and the focus is a locus of $\mathfrak{D}$ that is not part of a cut, a commutation occurs and the process is relaunched for the subdesigns of $\mathfrak{D}$ accessible from that action;
- otherwise $\mathfrak{D}$ has no left part and its focus σ is part of a cut with another design with last rule $(\sigma, \mathcal{N})$ – where the *directory* (N) collects the ramifications of the actions on the same focus σ. In this case:
  - if $I \notin \mathcal{N}$, then interaction fails;
  - otherwise, the reduction continues on the connected parts of the subdesigns obtained considering only the part I of $\mathcal{N}$.

Therefore, cut reduction either fails, or it never ends, or it ends up with a design that reduces to ✠. In this last case the cut net is said to be closed and no commutation step occurs during the reduction. When the cut reduction between two designs reduces to ✠, those designs are said to be orthogonal (to each other) – and can be noted $\mathfrak{D}$ and $\mathfrak{D}^{\perp}$.

The notion of *behaviour* is built on that of orthogonal designs: a behaviour is a set of designs equal to its bi-orthogonal. That is, all its designs successfully normalize with all the same counter-designs (i.e. orthogonal designs). It is important to signal that cut reduction is usually explicitly considered, within the frame of Ludics, as the natural form of *interaction* proper to logical proofs. Indeed, thanks to the enlargement of the universe of logical proofs by taking into account also “wrong” proofs (the paraproofs), the cut reduction appears now as the process by means of which the correct proofs can be identified as those leading to successful normalization. Actually, since a cut reduction, as an interaction between (para)proofs, involves at least two designs it is not yet possible to decide which one represent the correct proof (the one that Frege and the tradition of truth-valued semantics would call the proof of a truth). The criterion, wholly internal to Ludics, is that the “winning” design does not play the daimon during the interaction[1], so that this allows finally to identify correct proofs. But then, what is that gets proved by a design? Behaviours are the answer, as far as they denote formulas. Indeed, a set of designs equal to its bi-orthogonal contains all the correct proofs that prove a same formula – since they lead to successful interaction (cut reduction) with the same set of counter-designs.

Moreover, like a chronicle “walks” a linear *path* over a design, from the base up to some leaf, possibly a daimon, enumerating the actions of a sequence, so we have an interactive correspondent of the chronicle in the *dispute*. A *dispute* can be seen as the matching of two paths over two orthogonal designs. In particular, the dispute appears as a travel on the different chronicles, where actions in a chronicle are matched by counter-actions in the other chronicle, and the dispute stitches the chronicles together by jumping from one to the other on negative actions (see figure 6.1 on page 192 for an immediate explanation), up to the moment when one of the two chronicles introduces the daimon, thus stopping the interaction – and indicating the other chronicle’s design as the real proof.

[1]Typically at the end since it stops the interaction.

Bibliography

– Vito Michele Abrusci, Marco Romano, and Christophe Fouqueré. Knowledge Representation through Coherence Spaces. In Jan L.G. Dietz, editor, *proceedings of KEOD 2009, 1st International Conference on Knowledge Engineering and Ontology Developmnet*, pages 220–225. INSTICC - Institute for Systems and Technologies of Information, Control and Communication, 2009. ISBN 978-989-674-012-2.

– Vito Michele Abrusci, Marco Romano, and Christophe Fouqueré. Ontologies and Coherence Spaces. In *Ludics, Dialogue and Interaction*, volume 6505 of *FoLLI LNAI*, pages 205–219. Springer-Verlag, 2011.

– J.-M. Andreoli. Logic Programming with Focusing Proofs in Linear Logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.

– Aristotle. *"De Interpretatione"*. University of Virginia Library Electronic Text Center, 1928. Translated by E. M. Edghill.

– Aristotle. *The Categories*. Project Gutenberg, 2000. Translated by E. M. Edghill.

– F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logics Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

– Tim Berners-Lee. Issues for the future of the Web. Opening keynote at the First International World Wide Web Conference. `http://www.w3.org/2001/sw/Europe/graphics/tmbltext.html`, May 1994. Geneva.

– Tim Berners-Lee. The Semantic Web as a language of logic. `http://www.w3.org/DesignIssues/Logic.html`, 1999-2009.

– Tim Berners-Lee and Mark Fischetti. *Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. Harper San Francisco, 1999.

– Tim Berners-Lee, Dan Connolly, and alii. A PIM ontology. `http://www.w3.org/2000/10/swap/pim/contact`, 2000.

– Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, May 2001.

– Willem Nico Borst. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. PhD thesis, University of Twente, 1997.

– Paolo Bouquet, Themis Palpanas, Heiko Stoermer, and Massimiliano Vignolo. A Conceptual Model for a Web-Scale Entity Name System. In *ASWC*, pages 46–60, 2009.

– Christopher H. Brooks and Nancy Montanez. An analysis of the effectiveness of tagging in blogs. In *AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*. AAAI, 2005.

– Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez Muro, and Riccardo Rosati. Techniques for Ontology Interoperation (Deliverable 24). Technical report, TONES – Thinking ONtologiES, EU-FP6 project, 2006.

– Felice Cardone. The geometry and algebra of commitment. In *Ludics, Dialogue and Interaction*, volume 6505 of *FoLLI LNAI*, pages 59–79. Springer-Verlag, 2011.

– C. C. Chang and H. J. Keisler. *Model Theory*. North-Holland, 1976.

– Claudia Faggian. Travelling on Designs. In *Proceedings of the 16th International Workshop and 11th Annual Conference of the EACSL on Computer Science Logic*, CSL '02, pages 427–441, London, UK, 2002. Springer-Verlag.

– Christophe Fouqueré. Ludics and Web: Another Reading of Standard Operations. In *Ludics, Dialogue and Interaction*, volume 6505 of *FoLLI LNAI*. Springer-Verlag, 2011.

– Stan Franklin and Art Graesser. Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. In *Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages*, pages 21–35, London, UK, 1997. Springer-Verlag.

– Bernhard Ganter, Gerd Stumme, and Rudolf Wille, editors. *Formal Concept Analysis: Foundations and Applications*, volume 3626 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2005.

– Michael Genesereth and Nils J. Nilsson. *Logical foundations of Artificial Intelligence*. Morgan Kaufmann Publishers, Los Altos, California, 1987.

– Michael R. Genesereth and Steven P. Ketchpel. Software Agents. *Communications of the ACM*, 37:48–53, 1994.

– Silvio Ghilardi, Carsten Lutz, and Frank Wolter. Did I Damage my Ontology? A Case for Conservative Extensions in Description Logics. In *In Proceedings of KR2006*, pages 187–197. AAAI Press, 2006.

– Jean-Yves Girard. Linear Logic. *Theoretical Computer Science*, 50:1–102, 1987.

– Jean-Yves Girard. Locus solum: from the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, pages 301–506, 2001.

– Jean-Yves Girard. *Le Point Aveugle: Tome 1. Cours de Logique, Vers la perfection.* Hermann, July 2006.

– Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types.* Cambridge University Press, 1989.

– Bernardo Cuenca Grau, Carsten Lutz, Ulrike Sattler, and Anni-Yasmin Turhan. Tasks for Ontology Integration and Merging (Deliverable 11). Technical report, TONES – Thinking ONtologiES, EU-FP6 project, 2006.

– Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. A logical framework for modular integration of ontologies. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 298–303, 2007.

– Thomas Gruber. Ontology of Folksonomy: A Mash-up of Apples and Oranges. *International Journal On Semantic Web and Information Systems*, 3(1):1–11, 2007. available also online at `http://tomgruber.org/writing/ontology-of-folksonomy.htm`.

– Tom Gruber. Toward principles for the design of ontologies used for knowledge sharing. In *Formal Ontology in Conceptual Analysis and Knowledge Representation.* Kluwer Academic, 1993. Available as Stanford Knowledge Systems Laboratory Report KSL-93-04.

– Tom Gruber. Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human and Computer Studies*, 43(5-6):907–928, November 1995.

– Nicola Guarino. Formal Ontology, Conceptual Analysis and Knowledge Representation. *International Journal of Human and Computer Studies*, 43:625–640, 1995.

– Nicola Guarino. Formal Ontology and Information Systems. In *1st International Conference on Formal Ontologies in Information Systems, FOIS'98*, pages 3–15. IOS Press, Trento, Italy, 1998.

– Nicola Guarino, Massimiliano Carrara, and Pierdaniele Giaretta. Formalizing ontological commitments. In *AAAI'94: Proceedings of the 12th National conference on Artificial Intelligence*, volume 1, pages 560–567. American Association for Artificial Intelligence, 1994.

– Nicola Guarino, Daniel Oberle, and Steffen Staab. What is an Ontology? In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, pages 1–17. Springer Berlin Heidelberg, 2nd edition, 2009.

– P. Hayes. RDF Semantics. `http://www.w3.org/TR/2004/REC-rdf-mt-20040210/`, 2004.

– Jim Hendler. "a little semantics goes a long way". `http://www.cs.rpi.edu/~hendler/LittleSemanticsWeb.html`, 2009.

– Andreas Hotho, Robert Jaeschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, pages 411–426. Springer, 2006.

– Robert Jaschke, Andreas Hotho, Christoph Schmitz, Bernhard Ganter, and Gerd Stumme. Discovering shared conceptualizations in folksonomies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6:38–53, 2008.

– Yu-Kyung Kang, Suk-Hyung Hwang, and Kyoung-Mo Yang. FCA-based conceptual knowledge discovery in Folksonomy. In *World Academy of Science, Engineering and Technology*, volume 53, pages 842–846. 2009.

– Immanuel Kant. *Lectures on Metaphysics*. Cambridge University Press, 1997. Translated by Ameriks Karl and Naragon Steve.

– Alain Lecomte and Myriam Quatrini. Ludics and Its Applications to Natural Language Semantics. In *Logic, Language, Information and Computation. 16th International Workshop, WoLLIC 2009. Proceedings*, pages 242–255. Springer Berlin Heidelberg, 2009.

– Domenico Lembo, Carsten Lutz, and Boontawee Suntisrivaraporn. Tasks for Ontology Design and Maintenance (Deliverable 5). Technical report, TONES – Thinking ONtologiES, EU-FP6 project, 2006.

– Maurizio Lenzerini. Data integration: A theoretical perspective. In *PODS'02: Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of database systems*, pages 233–246. ACM Press, New York, 2002.

– Obitko Marek. `http://www.obitko.com/tutorials/ontologies-semantic-web/operations-on-ontologies.html`, 2006.

– Peter McBrien and Alexandra Poulovassilis. P2p query reformulation over both-as-view data transformation rules. In Gianluca Moro, Sonia Bergamaschi, Sam Joseph, Jean-Henry Morin, and Aris Ouksel, editors, *Databases, Information Systems, and Peer-to-Peer Computing*, volume 4125 of *Lecture Notes in Computer Science*, pages 310–322. Springer Berlin / Heidelberg, 2007.

– Peter Mika. Ontologies are us: A unified model of social networks and semantics. In *Web Semantics: Science, Services and Agents on the World Wide Web*, volume 5, pages 5–15. 2007.

– Zoltán Miklós, Nicolas Bonvin, Paolo Bouquet, Michele Catasta, Daniele Cordioli, Peter Fankhauser, Julien Gaugaz, Ekaterini Ioannou, Hristo Koshutanski, and Antonio Maña. From Web Data to Entities and Back. In *CAiSE*, pages 302–316, 2010.

– Alexandre Monnin. Qu'est-ce qu'un tag? Entre accès et libellés, l'esquisse d'une caractérisation. In *Actes des 20es Journées Francophones d'Ingénierie des Connaissances "Connaissance et communautés en ligne" (IC2009)*, pages 1–12. 2009.

– Kevin Mulligan. Métaphysique et ontologie. In Engel P., editor, *Précis de Philosophie analytique*, pages 5–33. Presses Universitaires de France, 2000.

– Frédéric Nef. *Logique, langage et réalité.* Editions Universitaires, 1991.

– Frédéric Nef. *Traité d'ontologie pour les non-philosophes (et les philosophes).* Folio essais, 2009.

– Natalya Noy and Carole Hafner. State of the art in the Ontology Design. *AI Magazine*, 18(3):53, 1997.

– Natalya Fridman Noy and Mark A. Musen. An algorithm for merging and aligning ontologies: Automation and tool support. In *In Proceedings of the Workshop on Ontology Management at the Sixteenth National Conference on Artificial Intelligence (AAAI-99).* AAAI Press, 1999.

– C. K. Ogden and I. A. Richards. *The Meaning of Meaning.* New York, Harcourt, Brace & World, Inc., 8th ed. edition, 1923.

– Jeff Z. Pan, Stuart Taylor, and Edward Thomas. Reducing Ambiguity in Tagging Systems with Folksonomy Search Expansion. In *ESWC 2009 Heraklion: Proceedings of the 6th European Semantic Web Conference on The Semantic Web*, pages 669–683, Berlin, Heidelberg, 2009. Springer-Verlag.

– C.I. Petrie. Agent-based engineering, the Web, and Intelligence. *IEEE Expert*, 11(6): 24–29, 1996.

– Lévy Pierre. *L'intelligence collective. Pour une anthropologie du cyberespace.* La Découverte, Paris, 1994.

– Roberto Poli. *Ontologia formale.* Marietti, Genova, 1992.

– Roberto Poli. Descriptive, formal and formalized ontologies. In Denis Fisette, editor, *Husserl's Logical Investigations Reconsidered*, pages 183–210. Kluwer Academic Publisher, 2003.

– Willard Van Ormond Quine. On What There Is. *"Review of Metaphysics"*, 2:21–38, 1948.

– Emanuele Quintarelli. `http://www.iskoi.org/doc/folksonomies.htm`, 2005. ISKO Italy-UniMIB meeting.

– Atefeh Sharif. Combining ontology and folksonomy: An Integrated Approach to Knowledge Representation. `http://www.ifla2009satelliteflorence.it/meeting3/program/assets/AtefehSharif.pdf`, 2009. IFLA satellite meetings. Florence, Italy.

– Barry Smith. Formal ontology, common sense and cognitive science. *International Journal of Human-Computer Studies*, 43:641–667, 1995.

– Barry Smith. Truthmaker Realism. *Australasian Journal of Philosophy*, 77(3):274–291, 1999.

– Barry Smith. Ontology. In Floridi L., editor, *Blackwell Guide to the Philosophy of Computing and Information*, pages 155–166. Blackwell, 2003. Oxford.

– Barry Smith. Beyond Concepts: Ontology as Reality Representation. In Achille Varzi and Laure Vieu, editors, *Formal Ontology and Information Systems. Proceedings of the Third International Conference (FOIS 2004)*, pages 73–84. IOS Press, Torino, Italy, 2004.

– Barry Smith. Against Fantology. In Johann C. Marek and Maria E. Reicher, editors, *Experience and Analysis*, pages 153–170. HPT&ÖBV, 2005.

– Barry Smith. Ontologia e sistemi informativi. In *Networks*, volume 6, pages 137–164. Servizio Web Italiano per la Filosofia, 2006.

– Barry Smith and Christopher Welty. FOIS introduction: Ontology – towards a new synthesis. In *Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01, pages .3–.9, New York, NY, USA, 2001. ACM.

– Steffen Staab and Rudi Studer, editors. *Handbook on Ontologies*. Springer Berlin Heidelberg, 2nd edition, 2009.

– Rudi Studer, V. Richard Benjamins, and Dieter Fensel. Knowledge Engineering: Principles and methods. In *Data & Knowledge Engineering*, volume 25, pages 161–197. Elsevier Science, 1998.

– Paul Studtmann. Aristotle's Categories. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. 2008. `http://plato.stanford.edu/archives/fall2008/entries/aristotle-categories`.

– Thomas Vander-Wal. Folksonomy. `http://vanderwal.net/folksonomy.html`, 2004.

– Thomas Vander-Wal. Explaining and Showing Broad and Narrow Folksonomies. `http://www.personalinfocloud.com/2005/02/explaining_and_.html`, 2005.

– C. Welty, F. Lehmann, G. Gruninger, and M. Uschold. Ontology: Expert Systems All Over Again? Invited panel at AAAI-99: The National Conference on Artificial Intelligence, 1999. Austin, Texas.

Acknowledgments

To write this thesis has been a very long work – I really would not have expected *so* long. Thus it has required not only help and support from all those who have been next to me during these intense months of writing, but it also has proven their patience.

First of all I want to thank my supervisors: Michele Abrusci at Roma Tre, who has been an endless source of hints and suggestions on almost all themes I've touched in this thesis – and yet other more. He has been always willing to discuss my doubts and perplexities about critical points in the thesis. His demand for precision and care for details in properly writing the most markedly logical parts of this work – though they made me feel chapter 4 and 5 a little tantalising – finally make this work to look much more like a logician's one. And Christophe Fouqueré at Paris 13, for his tremendous support in every aspect of my Ph.D., ranging from the 'personal' course in Ludics, to the provision of key ideas for its use in the Web, as well as of precious hints about Latex (and pieces of code too); for his impressively fast reviews – with proof-reading too – of papers and chapters of the thesis, and the invaluable help with French university bureaucracy. Together, my supervisors have incomparably helped me to properly set questions and issues emerging from the research, and to solve them. By the way, our meetings in Italy and France have been very nice occasions to put hard work together with fine cuisine and pleasant informal discussions.

My gratitude goes in particular to Luca Severini, who made it all possible: firstly by introducing me to ontologies during my initial training at Epistematica – long time ago I can say now – and especially during the Ph.D. course by being the sponsor of my grant, by staying always present looking after the development of my thesis, and continuously showing his trust in me.

My thanks to the Gruppo di Logica at Roma Tre: Lorenzo, Roberto, Teresa, Marco for their interest in my research and their useful suggestions and encouragement; to my colleague Giulia for our attempts to study Linear Logic together and for the sharing of the 'fears' of doctoral students; to Paolo 'the elder', my mentor firstly at Roma Tre and then also at Paris 13, and a true friend; to Paolo 'the younger' for his help to keep the website up-and-running; to Giulio and Andrei too, my 'younger colleagues', and all the people I've shared my time with in our *stanzetta dottorandi*.

My thanks to the *équipe LCR* at Paris 13 and all the people who received me during my stay at the LIPN, especially Rushed, Sophie, Vincent, Luca and Daniele, and the whole 'Italian community' there; and to Jonathan for our discussions about folksonomies.

Special thanks to the secretary staff both at Roma Tre – especially Luciana, Concetta, Attilio and Andrea – for their help in setting up the cotutelle and during my period as *Rappresentante dei dottorandi*, and at Paris 13 – especially Brigitte.

Finally, my greatest thank to my family. Especially to my parents for their constant support and their patience with me especially during the writing of the thesis; to my brother Gian Francesco for his tolerance for my nightly work in our room; to Alessandro and Silvia who have always encouraged me and to my little lovely nieces for their contagious cheerfulness. And to all of my friends, many of which are still waiting to see me back from the thesis writing.