# ICT4LAW
# Articoli che presentano i risultati del progetto

## INDICE

# Legal Taxonomy Syllabus version 2.0

Gianmaria Ajani[1], Guido Boella[2], Leonardo Lesmo[2], Marco Martin[2], Alessandro Mazzei[2], Daniele P. Radicioni[2], and Piercarlo Rossi[3]

[1] Dipartimento di Scienze Giuridiche - Università di Torino
[2] Dipartimento di Informatica - Università di Torino
[3] Dipartimento di Studi per l'Impresa e il Territorio - Università del Piemonte Orientale

gianmaria.ajani@unito.it, {guido,lesmo,mazzei,radicion}@di.unito.it, notmart@gmail.com, piercarlo.rossi@eco.unipmn.it

**Abstract.** The need for managing the conceptual representation of European law led to the development of the Legal Taxonomy Syllabus (LTS) and the related methodology. In this paper we consider further legal issues that emerged during the test and use phases, and outline the new features that we added to the new version, the LTS 2.0.

## 1 Introduction

European Union Directives (EUDs) are sets of norms that have to be implemented by the national legislations and translated into the language of each Member State. The general problem of multilingualism in European legislation has recently been tackled by linguistic and ontological tools [8,5,17,18]. The management of EUD is particularly complex, since the *implementation* of a EUD does not correspond to a straight transposition into a national law.

In previous work we carried out the Legal Taxonomy Syllabus[4] (LTS), a tool to build multilingual conceptual dictionaries aimed at representing and analysing terminologies and concepts from EUDs [1,2]. LTS is based on the distinction between *terms* and *concepts*. The latter ones are arranged into ontologies that are organised in levels. Only two levels were defined: the European level –containing only one ontology deriving from EUDs annotations–, and the national level –hosting the distinct ontologies deriving from the legislations of EU member states.

While annotating the EUDs, testing and using the system, some more requirements emerged from users expert in law, demanding for a more sophisticated approach along with further developmental efforts: first, it is frequent the case of concepts which are the result of a doctrinal interpretation process rather than of the definition in directives. If, on the one hand, the definitions in directives and their relation with the actual text are required by legal scholars to have a precise model of European law, the layman is more interested in the

[4] http://www.eulawtaxonomy.org

concepts which results from the doctrinal interpretation. Furthermore, laws are typical objects evolving through time. An open issue to cope with in building legal frameworks both at the European and at the national level is the *normative change* [12,4]. Concepts in the legal ontologies should not only represent the consolidated legal text, but should also keep trace of the evolution of meaning.

In this paper we consider not only the terms defined in the directives, but also the interpretation process of legal scholars in the LTSand how to better integrate concepts and the text of EUDs in the LTS. We answer the first question by introducing *abstract* concepts (*abstract* in that they are not related to a single directive), which should be conveniently recognized as a *grouping* of concepts. The users will be thereby allowed to navigate the ontology at different levels of details depending on their goals. Moreover, exploiting natural language processing techniques we greatly simplify the management of legal text associated to concepts. Also, we investigate how to extend the ontology with a temporal dimension to the ends of representing *normative change*, and to allow users to search also for past meanings of terms and the modified norms introducing them. To these ends, we introduce *time* into the ontology, and allow new concepts to replace the old ones while keeping the latter ones in the system as well.

## 2 Multilingual and Multilevel Ontologies for European Directives

Comparative Law has identified two key points in dealing with EUD, which make more difficult dealing with the polysemy of legal terms. We call them the *terminological* and *conceptual misalignments*. The first problem is determined by the lexical ambiguity of the legal terms (in particular homonymy) in the translation of EUDs. The second problem is determined by the lexical and conceptual ambiguity of the legal terms (in particular polysemy) in the implementation of EUDs. These issues determined the development of the first release of the LTS, and have been illustrated in [2].

We now illustrate further issues in handling EUDs that required to devise further features to enrich the original LTS.

### 2.1 Concepts Abstraction

The LTS system relies on the concept of *unitary-meaning* or *umeaning*: such atomic concepts can be derived from excerpts of the text of legal norms, such as European directives or national laws, and are arranged into two separate categories of umeanings, as described in [2]. EUDs provide rigorous definitions of some terms, such as the definition of the Italian term *consumatore* (*consumer*), in the Italian version of the *EUD 93/13/EEC*, Art. 2 is:

> [. . . ](b) “consumatore”: qualsiasi persona fisica che, nei contratti oggetto della presente direttiva, agisce per fini che non rientrano nel quadro della sua attività professionale; [...]

> [. . . ](b) "consumer": means any natural person who, in contracts covered by this Directive, is acting for purposes which are outside his professional activity; [. . . ] (*our literal translation*)

However, two facts must be pointed out. Different EUDs might affect different aspects of the legislation: thus the definition of a term in a EUD only applies to a specific context. Furthermore, EUDs could be written at different points in time, and they can introduce diverging definitions. Let us consider the definition of *consumatore*, as it appears in the Italian version of the *EUD 2002/65/EC, Art. 1*:

> [. . . ](d) "consumatore": qualunque persona fisica che, nei contratti a distanza, agisca per fini che non rientrano nel quadro della propria attività commerciale o professionale; [. . . ]
>
> [. . . ](d) "consumer": means any natural person who, in distance contracts covered by this Directive, is acting for purposes which are outside his business or professional activity; [. . . ] (*our literal translation*)

We remark that in contrast with English, in Italian the second definition of *consumatore* is broader than the first one, since the term *professionale* (*professional*) does not include *commerciale* (*business*). This divergence of term definitions can often occur, since EUDs have usually a sectorial specific target. In this way, EUDs covering different sectors can provide different definitions, and as many views on the same concept. Lawyers and legislators started to put together highly sectorial concepts into more abstract concepts with broader meaning, in order to describe (complex) entities, such as the *consumatore* in all of its aspects.

In recent years, in the Italian legislation EUDs are not being implemented as single laws, but rather as groups of EUDs. The juridical concepts are defined as the union of all the sectorial concepts provided by the individual EUDs, as a result of the doctrinal interpretation process of directives. These problems are common to all European languages. Consider, for instance the definition of *consumer*, in the English version of the EUD 1999/44/EC, Art. 1.2 is:

> [. . . ] (a) consumer: shall mean any natural person who, in the contracts covered by this Directive, is acting for purposes which are not related to his trade, business or profession; [. . . ]

that has a different meaning with respect to the definition of *consumer* given in the Council Directive 90/314/EEC, Art. 2.4:

> [. . . ] "consumer" means the person who takes or agrees to take the package ('the principal contractor'), or any person on whose behalf the principal contractor agrees to purchase the package ('the other beneficiaries') or any person to whom the principal contractor or any of the other beneficiaries transfers the package ('the transferee') [. . . ]

The LTS should be able to represent both the more specific dimension related to the definitions in EUDs and the more abstract one which results from the doctrinal interpretation of European law. The LTS allows inserting the text paragraphs where *umeanings* are defined. However, to gain better understanding of legal concepts, it is often required to consider a broader fragment. For example, in the case of *consumer* the definition is not enough, and it is necessary to collect multiple paragraphs where consumer protection norms are presented and discussed.

### 2.2 Normative Change

Another big open issue to cope with in building tools for describing legal frameworks both at the European and at the national level is the *normative change* [12]. One major problem, well-known in the literature, is the update of *non-monotonic* ontologies and knowledge bases [4]. In other words, not necessarily ontologies and knowledge bases have a structure constant through time (e.g., see [16]): concepts and relations present in the ontology can become obsolete as new concepts and relations are added. This is indeed the case of legal frameworks, that are continuously modified as new laws can modify paragraphs of old ones.

We can have two types of normative change: *explicit* change and *implicit* change. In the first case the new norm explicitly states the abrogation of a specific paragraph of an old law (for details on this line of investigation, please refer to [6]). Alternatively, the newer law can state a concept in contradiction to previous laws, but without mentioning them explicitly. In this case the concept stated by the new law becomes the current one; also, the parts of the old laws affected by changes (no longer updated) become obsolete.

## 3 From LTS 1.0 to LTS 2.0

In this Section we first summarize the functionalities of the existing LTS [2], and then we explain how it has been extended to cope with the new requirements described in the previous Section.

### 3.1 LTS 1.0

The main assumptions of our methodology come from studies in comparative law [13] and ontologies engineering [10]. Terms –*lexical entries* for legal information–, and concepts must be distinguished; for this purpose we use lightweight ontologies, i.e. simple taxonomic structures of primitive or composite terms together with associated definitions. They are hardly axiomatized as the intended meaning of the terms used by the community is more or less known in advance by all members, and the ontology can be limited to those structural relationships among terms that are considered as relevant.

We distinguish the ontology implicitly defined by EUD, the *EU level*, from the various national ontologies. Each one of these "particular" ontologies belongs to the *national level*: i.e., each national legislation refers to a distinct national legal ontology. We do not assume that the transposition of an EUD automatically

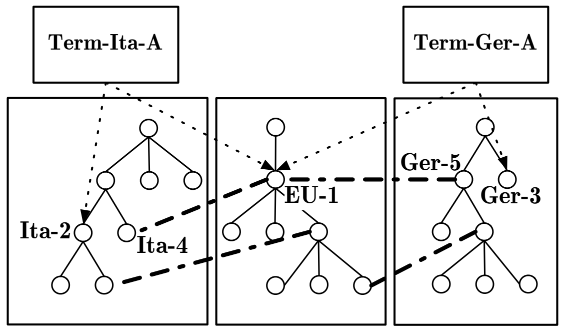


**Fig. 1.** Relationship between ontologies and terms. The thick arcs represent the inter-ontology "association" link.

introduces in a national ontology the same concepts that are present at the EU level.

Corresponding concepts at the EU level and at the national level can be denoted by different terms in the same national language.

A standard way to properly manage large multilingual lexical databases is to make a clear distinction among terms and their interlingual acceptions (or *axies*) [15].

In the LTS project to properly manage terminological and conceptual misalignment, we distinguish the notion of *legal term* from the notion of *legal concept* and we build a systematic classification based on this distinction. The basic idea in our system is that the conceptual backbone consists in a taxonomy of concepts (ontology) to which the terms can refer in order to express their meaning. One of the main points to keep in mind is that we do not assume the existence of a single taxonomy covering all languages. In fact, the different national systems may organize the concepts in different ways. For instance, the term *contract* corresponds to different concepts in common law and civil law, where it has the meaning of *bargain* and *agreement*, respectively [14]. In most complex instances, there are no homologous between terms-concepts such as *frutto civile* (legal fruit) and *income*, but respectively civil law and common law systems can achieve functionally similar operational rules thanks to the functioning of the entire taxonomy of national legal concepts [9]. Consequently, the LTS includes different ontologies, one for each involved national language plus one for the language of EU documents. Each language-specific ontology is related via a set of *association* links to the EU concepts, as shown in Fig. 1.

Although this picture is conform to intuition, in the basic LTS it has been implemented by taking two issues into account. First, it must be observed that the various national ontologies have a reference language. This is not the case for the EU ontology. For instance, a given term in English could refer either to a concept in the UK ontology or to a concept in the EU ontology. In the first case, the term is used for referring to a concept in the national UK legal system,

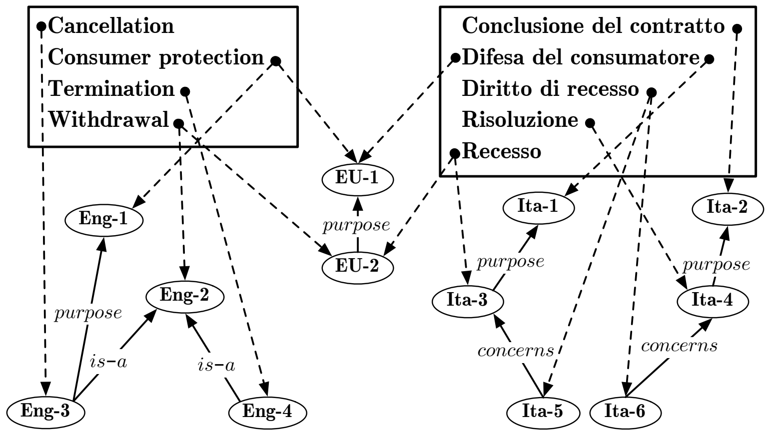


**Fig. 2.** An example of interconnections among terms.

whilst in the second one, it is used to refer to a concept used in the European directives. This is one of the main advantages of LTS. For example *klar und verständlich* could refer both to concept `Ger-379` (a concept in the German Ontology) and to concept `EU-882` (a concept in the European ontology). This is the LTS solution for facing the possibility of a partial correspondence between the meaning of a term in the national system and the meaning of the same term in the translation of a EU directive. This feature enables the LTS to be more precise about what "translation" means. It makes available a way for asserting that two terms are the translation of each other, but just in case those terms have been used in the translation of an EU directive: within LTS, we can talk about direct EU-to-national translations of terms, and about *implicit* national-to-national translations of terms. In other words, we distinguish between *explicit* and *implicit* associations among concepts belonging to different levels. The former ones are direct links that are explicitly used by legal experts to mark a relation between concepts. The latter ones are indirect links: if we start from a concept at a given national level, by following a direct link we reach another concept at European level. Then, we will be able to see how that concept is mapped onto further concepts at the various national levels.

The situation enforced in LTS is depicted in Fig. 1, where it is represented that the Italian term *Term-Ita-A* and the German term *Term-Ger-A* have been used as corresponding terms in the translation of an EU directive, as shown by the fact that both of them refer to the same EU-concept `EU-1`. In the Italian legal system, *Term-Ita-A* has the meaning `Ita-2`. In the German legal system, *Term-Ger-A* has the meaning `Ger-3`. The EU translations of the directive is correct insofar no terms exist in Italian and German that characterize precisely the concept `EU-1` in the two languages (i.e., the "associated" concepts `Ita-4` and `Ger-5` have no corresponding legal terms). A practical example of such a situa-

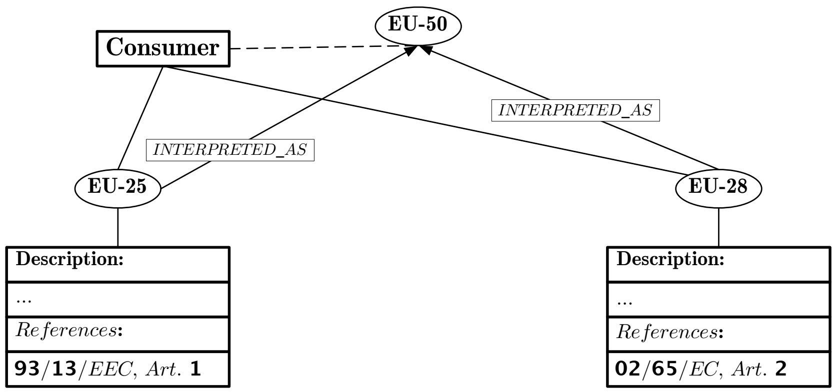


**Fig. 3.** Umeanings `Eu-25` and `Eu-28` are interpreted by the more abstract umeaning `Eu-50`, the link between `Eu-50` and the term "consumer" is implicit.

tion is reported in Fig. 2, where we can see that the ontologies include different types of arcs. Beyond the usual *is-a* (linking a category to its supercategory), there are also the arcs *purpose*, which relate a concept to the legal principle motivating it, and *concerns*, which refer to a general relatedness. The dotted arcs represent the reference from terms to concepts. Some terms have links both to a National ontology and to the EU Ontology (in particular, *withdrawal* vs. *recesso* and *difesa del consumatore* vs. *consumer protection*).

The last item above is especially relevant: note that this configuration of arcs specifies that: 1) *withdrawal* and *recesso* have been used as equivalent terms (concept `EU-2`) in some European Directives (e.g., Directive 90/314/EEC). 2) In that context, the term involved an act having as purpose some kind of protection of the consumer. 3) The terms used for referring to the latter are *consumer protection* in English and *difesa del consumatore* in Italian. 4) In the British legal system, however, not all *withdrawals* have this goal, but only a subtype of them, to which the code refers to as *cancellation* (concept `Eng-3`). 5) In the Italian legal system, the term *diritto di recesso* is ambiguous, since it can be used with reference either to something concerning the *risoluzione* (concept `Ita-4`), or to something concerning the *recesso* proper (concept `Ita-3`).

### 3.2 Enhancing LTS with interpretation and abstraction

As described in Section 2.1, different pieces of legislations can bear different definitions of terms. Having different detailed definitions is important during the interpretation of very sectorial legal cases, but for the general case it is important to have a view that abstracts from the peculiarities of specific domains.

In order to solve this problem we introduced a new kind of ontologic relation called *INTERPRETED_AS*: it is a non transitive relation where the more general umeaning, that we will call *group leader* represents the abstracted concept that groups the meaning of a number of more specific umeanings, that are the sectorial umeanings defined in the individual EUDs or national laws (see Fig. 3).

We have also introduced a number of constraints and integrity checks to ensure that the semantics of the grouping concept is respected and to improve the usability of the system: *i*) each umeaning can belong to a single group; *ii*) a group leader cannot exist without group members; *iii*) when the user searches into the umeaning database, more specific umeanings are excluded from the results unless the user explicitly asks to show them, i.e. only the group leaders are shown in the results. The need to contextualize concepts to the EUDs defining them leads to the need of more complex instruments to deal with the language of the norms. An umeaning is defined by the legal texts themselves; this makes clear that the creation of umeaning is a quite long task, because it requires from the user searching and reading a very large number of documents.

In order to ease this process, we developed a database that contains the full versions of the desired EUDs and national laws. In this way, the user can carry out his task according to the following workflow. *1*) The user creates a new *umeaning* linked with the term he wants to define; *2*) He selects relevant citation from legal text; consequently, the browser is redirected to a search page and the main term attached to the umeaning is used as the default query; *3*) After choosing one of the search results, the full text of the legal document is displayed, with the search terms highlighted; *4*) Finally, the user selects the text that will go in the citation with the mouse and confirms the insertion in the references database. Lastly, when the user searches for a term in the documents database, the search is not performed upon the exact words, rather with their roots, so for instance when performing the search on the term "contracts" also documents containing only "contract" will be found, this seems to enhance the information retrieval performance as shown in [11].

### 3.3 LTS with normative change

When a new normative is approved and enacted it can define a number of new umeanings; moreover it can happen that the same law can change a number of old umeanings defined by old laws. In particular, these old umeaning can become obsolete and no longer valid. We are aware of the difficulties concerning the modelling of the time in artificial intelligence and in formal ontology too[5]. Anyway, in LTS we adopted a naive solution in order to manage the simpler situation concerning t In the LTS it was necessary to delete all old umeanings, causing the loss of all historic informations from the database, informations that are quite valuable to better understand the evolving of the normative. This problem was resolved by using the same solution adopted for the interpretation and abstraction of the norms (Section 3.2), i.e. empowering LTS with a new ontological relation called *REPLACED_BY*.

When the paragraph of an EUD defining an umeaning has been modified by a new EUD, the new one defines a new umeaning that will replace the old umean-

[5] E.g. see [3] for a general survey and [12] for normative systems

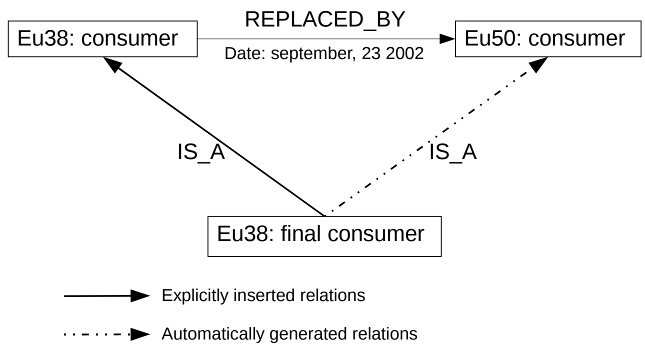


**Fig. 4.** An example of use of the *REPLACED_BY* relation.

ing in the ontology. There will be a relation of type *REPLACED_BY* between the two umeanings, where the child umeaning is replaced by the more general umeaning. Also in this case the new ontological relation has some peculiar characteristics that distinguishes it from the usual ontological relations (Figure 4): *i*) a *REPLACED_BY* relation brings with it a new data field not present in the other relations: the substitution date; *ii*) when the user performs a search in the umeanings database the replaced ones will not be shown, unless the user asks for a certain past date, thus obtaining a snapshot of the legal ontology that was valid in that particular moment; *iii*) when a new umeaning replaces an old one all the ontological relations where the old umeaning appeared are automatically copied in the new umeaning. If some of them are no longer valid with the new umeaning, manual intervention from the user is required.

## 4 Conclusions

In this paper we discuss some features that have recently been introduced in the LTS, a tool for building multilingual conceptual dictionaries for the EU law. The tool is based on lightweight ontologies to emphasize the distinction between concepts and terms. Different ontologies are built at the EU level and for each national language, to deal with polysemy and terminological and conceptual misalignment.

The present work illustrates how to distinguish between concepts as they are defined in the text of the directives and the concepts representing the doctrinal interpretation of the terms. Moreover, we point out how to deal with normative change by introducing a temporal dimension in ontologies.

Future work will involve exploring how to extend the LTS ontology, with special focus on the issue of populating it at the various levels by semi-automatic approaches [7].

## References


1. G. Ajani, G. Boella, L. Lesmo, M. Martin, A. Mazzei, and P. Rossi. A development tool for multilingual ontology-based conceptual dictionaries. In *Proc. of 5th International Conference on Language Resources and Evaluation, LREC06*, pages 1–6, Genoa, 2006.
2. G. Ajani, G. Boella, L. Lesmo, A. Mazzei, and P. Rossi. Terminological and ontological analysis of european directives: multilinguism in law. In *11th Internation Conference o Arificial Intelligence and Law (ICAIL)*, pages 43–48, 2007.
3. J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.
4. M. Cadoli and F. M. Donini. A Survey on Knowledge Compilation. *AI Communications*, 10(3–4):137–150, 1997.
5. P. Casanovas, N. Casellas, C. Tempich, D. Vrandecic, and R. Benjamins. OPJK modeling methodology. In *Proceedings of the ICAIL Workshop: LOAIT 2005*, 2005.
6. M. Cherubini, G. Giardiello, S. Marchi, S. Montemagni, P. Spinosa, and G. Venturi. NLP-based metadata annotation of textual amendments. In *Proc. of WORKSHOP ON LEGISLATIVE XML 2008, JURIX 2008*, 2008.
7. P. Cimiano. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer, 2006.
8. S. Després and S. Szulman. Merging of legal micro-ontologies from european directives. *Journal of Artificial Intelligence and Law*, February 2007.
9. M. Graziadei. Tuttifrutti. In P. Birks and A. Pretto, editors, *Themes in Comparative Law*, pages –. Oxford University Press, 2004.
10. M. Klein. Combining and relating ontologies: an analysis of problems and solutions. In *Workshop on Ontologies and Information Sharing, IJCAI'01*, Seattle, USA, 2001.
11. R. J. Krovetz. *Word sense disambiguation for large text databases*. PhD thesis, University of Massachusetts, 1995.
12. M. Palmirani and R. Brighi. Time model for managing the dynamic of normative system. *Electronic Government*, pages 207–218, 2006.
13. P. Rossi and C. Vogel. Terms and concepts; towards a syllabus for european private law. *European Review of Private Law (ERPL)*, 12(2):293–300, 2004.
14. R. Sacco. Contract. *European Review of Private Law*, 2:237–240, 1999.
15. G. Sérasset. Interlingual lexical organization for multilingual lexical databases in NADIA. In *Proc. COLING94*, pages 278–282, 1994.
16. The Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. *Nature Genetics* *http://genetics.nature.com*, 25:25–29, 2000.
17. Daniela Tiscornia. The Lois Project: Lexical Ontologies for Legal Information. In *Proceedings of the V Legislative XML Workshop*. European Press Academic Publishing, 2007.
18. P. Vossen, W. Peters, and J. Gonzalo. Towards a universal index of meaning. In *Proc. ACL-99 Siglex Workshop*, 1999.

# Multilevel Legal Ontologies

Gianmaria Ajani[1], Guido Boella[2], Leonardo Lesmo[2], Marco Martin[2], Alessandro Mazzei[2], Daniele P. Radicioni[2], and Piercarlo Rossi[3]

[1] Dipartimento di Scienze Giuridiche - Università di Torino
[2] Dipartimento di Informatica - Università di Torino,
[3] Dipartimento di Studi per l'Impresa e il Territorio, Università del Piemonte Orientale
gianmaria.ajani@unito.it,
{guido,lesmo,mazzei,radicion}@di.unito.it,
notmart@gmail.com,
piercarlo.rossi@eco.unipmn.it

February 13, 2009

**Abstract.** In order to manage the conceptual representation of European law we have proposed the Legal Taxonomy Syllabus (LTS) and the related methodology. In this paper we consider further legal issues that emerged during the testing and use of the LTS, and how we take them into account in the new release of the system. In particular, we consider the problem of representing interpretation of terms besides the definitions occurring in the directives, the problem of normative change and the process of planning legal reforms of European law. In particular, we show how to include into the Legal Taxonomy Syllabus the Acquis Principles - which have been sketched by scholars in European Private Law from the so-called Acquis communautaire -, how to take the temporal dimension into account in ontologies, and how to apply techniques of natural language processing to the legal texts uploaded in the LTS.

## 1 Introduction

European Union Directives (EUDs) are sets of norms that have to be implemented by the national legislations and translated into the language of each Member State. The general problem of multilingualism in European legislation has recently been addressed by using linguistic and ontological tools, e.g. [13,10,16,29,30]. The management of EUD is particularly complex, since the *implementation* of a EUD does not correspond to a straight transposition into a national law. By converse, managing this kind of complexity with appropriate tools can facilitate the comparison and harmonization of national legislation [7]. For instance, LOIS Project aims at extending EuroWordnet with legal information, thus adopting a similar approach to multilinguism, with the goal of connecting legal ontology to a higher level ontology [29].

In previous work we proposed the Legal Taxonomy Syllabus[4] (LTS), a tool to build multilingual conceptual dictionaries aimed at representing and analysing terminologies and concepts from EUDs [2,1,3,4]. In some respect, the target of LTS is similar to the one of the European project LOIS, but while the final goal of LOIS is to support applications concerning information extraction, the LTS is concerned with the access of human experts to the EU documents. LTS is based on the distinction between *terms* and *concepts*. The latter ones are arranged into ontologies that are organised in levels. In [4] only two levels were defined: the European level –containing only one ontology deriving from EUDs annotations–, and the national level –hosting the distinct ontologies deriving from the legislations of EU member states.

While annotating the EUDs, testing and using the system, some more requirements emerged from users expert in law, demanding for a more sophisticated approach along with further developmental efforts: first, we noted that it is frequent the case of concepts which are the result of a doctrinal interpretation process rather than of the definition in directives. If, on the one hand, the definitions in directives and their relation with the actual text are required by legal scholars to have a precise model of European law, the layman is more interested in the concepts which results from the doctrinal interpretation. Second, laws are typical objects evolving through time. A big open issue to cope with in building legal frameworks both at the European and at the national level is the *normative change* [23,9]. Concepts in the legal ontologies should not only represent the consolidated legal text, but should also keep trace of the evolution of meaning. Finally, besides the actual directives, the European Union aims to harmonize law by reformulating terminology in a more coherent way. The European Commission provide in various way common principles, terminology, and rules for law to address gaps, conflicts, and ambiguities emerging from the application of European law, and this effort should be taken into account in the LTS as well.

Thus, in this paper we address the following research questions:

- How to consider not only the terms defined in the directives but also the interpretation process of legal scholars in the LTS? How to better integrate concepts and the text of EUDs in the LTS?
- How to extend the ontology with a temporal dimension to be able to represent normative change? How to allow users to search also for past meanings of terms and the modified norms introducing them?
- How to extend the levels of the LTS from European and national to new ones representing the possible reforms of European law? How to represent the relation between the existing European law and the planned revisions?

We answer the first question by introducing concepts called *abstract*, in that they are not related to a single directive, which should be conveniently recognized

[4] LTS is a dictionary of Consumer Law, which has been carried out within the broader scope of the Uniform Terminology Project, `http://www.uniformterminology.unito.it` [24]. The implemented system can be found at the URL: `http://www.eulawtaxonomy.org`.

as a *grouping* of concepts. The users will be allowed to navigate the ontology at different levels of details depending on their goals. Moreover, we use natural language processing techniques to facilitate the management of legal text associated to concepts.

We answer the second question by introducing *time* into the ontology and by allowing to have new concepts replacing the old ones while keeping the latter in the system as well.

We answer the third question by introducing new levels in the LTS and show how can they manage a set of principles (namely, the *Acquis Principles*) which is gaining in popularity to the ends of ameliorating the quality of EUDs national implementations by Member States.

The paper is structured as follows. In Section 2 we introduce the new legal requirements we take into account in the revision of the LTS. In Section 3 we first summarize the LTS (Sec. 3.1) and then we explain how the new system satisfies the additional requirements previously outlined (Sections 3.2 to 3.4). Conclusions end the paper.

## 2 Multilingual and Multilevel Ontologies for European Directives

In this Section we start by briefly summarizing the motivations which lead to the development of the LTS (Section 2.1), and then we introduce new requirements which have been raised by legal experts using the LTS (Sections 2.2 to 2.4).

### 2.1 Terminological and conceptual misalignment

Comparative Law has identified two key points in dealing with EUD, which make more difficult dealing with the polysemy of legal terms: we call them the *terminological* and *conceptual misalignments*.

In the case of EUD (usually adopted for harmonising the laws in the Member States), the terminological matter is complicated by the need to implement them in the national legislations. In order to have a precise transposition in a national law, a Directive may be subject to further interpretation. A single concept in a particular language can be expressed in a number of different ways in a EUD and in the national law implementing it. As a consequence we have a terminological misalignment. For example, the concept corresponding to the word *reasonably* in English, is translated into Italian as *ragionevolmente* in the EUD, and as *con ordinaria diligenza* into the transposition law.

In the EUD transposition laws a further problem arises from the different national *legal doctrines*. A legal concept expressed in an EUD may not be present in a national legal system. In this case we can talk about a conceptual misalignment. To make sense for the national lawyers' expectancies, the European legal terms have not only to be translated into a sound national terminology, but they also need to be correctly detected when their meanings refer to EU legal concepts or when their meanings are similar to concepts which are known in the

Member states. Consequently, the transposition of European law in the parochial legal framework of each Member state can lead to a set of distinct national legal doctrines, that are all different from the European one. In the case of consumer contracts (like those concluded by the means of distance communication as in Directive 97/7/EC, Art. 4.2), a related example of this phenomenon concerns the notion of providing in a *clear and comprehensible manner* some elements of the contract by the professionals to the consumers represents a specification of the information duties which are a pivotal principle of EU law. Despite the pairs of translation in the language versions of EU Directives (e.g., *klar und verständlich* in German - *clear and comprehensible* in English - *chiaro e comprensibile* in Italian), each legal term, when transposed in the national legal orders, is influenced by the conceptual filters of the lawyers' domestic legal thinking. So, *klar und verständlich* in the German system is considered by the German commentators referring to three different legal concepts: 1) the print or the writing of the information must be clear and legible (*Gestaltung der Information*), 2) the information must be intelligible by the consumer (*Formulierung der Information*), 3) the language of the information must be the national of consumer (*Sprache der Information*). In Italy, the judiciary tend to control more the formal features of the concepts 1 and 3, and less concept 2, while in England the main role has been played by the concept 2, though considered as plain style of language (not legal technical jargon) thanks to the historical influences of plain English movement in that country.

Note that this kind of problems identified in comparative law has a direct correspondence in the ontology theory. In particular Klein [19] has remarked that two particular forms of ontology mismatch, are *terminological* and *conceptualization* ontological mismatch which straightforwardly correspond to our definitions of misalignments.

### 2.2 Concepts Abstraction

The LTS system relies on the concept of *unitary-meaning* or *umeaning*: such atomic concepts can be derived from excerpts of the text of legal norms, such as European directives or national laws, and are arranged into two separate categories of umeanings, as described in [4].

EUDs provide rigorous definitions of some terms, such as the definition of the Italian term *consumatore* (*consumer*), in the Italian version of the *EUD 93/13/EEC*, Art. 2 is:

> [...](b) "consumatore": qualsiasi persona fisica che, nei contratti oggetto della presente direttiva, agisce per fini che non rientrano nel quadro della sua attività professionale; [...]
>
> [...](b) "consumer": means any natural person who, in contracts covered by this Directive, is acting for purposes which are outside his professional activity; [...] (*our literal translation*)

However, two facts must be pointed out. Different EUDs might affect different aspects of the legislation: thus the definition of a term in a EUD only applies to a specific context. Furthermore, EUDs could be written at different points in time, and they can introduce diverging definitions. Let us consider the definition of *consumatore*, as it appears in the Italian version of the *EUD 2002/65/EC, Art. 1*:

> [...](d) "consumatore": qualunque persona fisica che, nei contratti a distanza, agisca per fini che non rientrano nel quadro della propria attività commerciale o professionale; [...]
>
> [...](d) "consumer": means any natural person who, in distance contracts covered by this Directive, is acting for purposes which are outside his business or professional activity; [...] (*our literal translation*)

We remark that in contrast with English, in Italian the second definition of *consumatore* is broader than the first one, since the term *professionale* (*professional*) does not include *commerciale* (*business*). This divergence of term definitions can often occur, since EUDs have usually a sectorial specific target. In this way, EUDs covering different sectors can provide different definitions, and as many views on the same concept. Lawyers and legislators started to put together highly sectorial concepts into more abstract concepts with broader meaning, in order to describe (complex) entities, such as the *consumatore* in all of its aspects.

In recent years, in the Italian legislation EUDs are not being implemented as single laws, but rather as groups of EUDs. The juridical concepts are defined as the union of all the sectorial concepts provided by the individual EUDs, as a result of the doctrinal interpretation process of directives.

We remark that these problems are common to all European languages. Consider, for instance the definition of *consumer*, in the English version of the EUD 1999/44/EC, Art. 1.2 is:

> [...] (a) consumer: shall mean any natural person who, in the contracts covered by this Directive, is acting for purposes which are not related to his trade, business or profession; [...]

that has a different meaning with respect to the definition of *consumer* given in the Council Directive 90/314/EEC, Art. 2.4:

> [...] "consumer" means the person who takes or agrees to take the package ('the principal contractor'), or any person on whose behalf the principal contractor agrees to purchase the package ('the other beneficiaries') or any person to whom the principal contractor or any of the other beneficiaries transfers the package ('the transferee') [...]

The LTS should be able to represent both the more specific dimension related to the definitions in EUDs and the more abstract one which results from the doctrinal interpretation of European law. The LTS allows inserting the text paragraphs where *umeanings* are defined. However, to gain better understanding of legal concepts, it is often required to consider a broader fragment. For example, in the case of *consumer* the definition is not enough, and it is necessary to collect multiple paragraphs where consumer protection norms are presented and discussed.

### 2.3 Normative Change

Another big open issue to cope with in building tools for describing legal frameworks both at the European and at the national level is the *normative change* [23]. One major problem, well-known in the literature, is the update of *non-monotonic* ontologies and knowledge bases [9]. In other words, not necessarily ontologies and knowledge bases have a structure constant through time (e.g., see [28]): concepts and relations present in the ontology can become obsolete as new concepts and relations are added. This is indeed the case of legal frameworks, that are continuously modified as new laws can modify paragraphs of old ones.

We can have two types of normative change: *explicit* change and *implicit* change. In the first case the new norm explicitly states the abrogation of a specific paragraph of an old law (for details on this line of investigation, please refer to [11,8]). Alternatively, the newer law can state a concept in contradiction to previous laws, but without mentioning them explicitly. In this case the concept stated by the new law becomes the current one; also, the parts of the old laws affected by changes (no longer updated) become obsolete.

### 2.4 Reforming European Law: toward a Common Frame of Reference

In February 2003, the European Commission adopted a further communication entitled "A More Coherent European Contract Law - An Action Plan" [14]. One of the key measures proposed in the Action Plan was the elaboration of a Common Frame of Reference (CFR). According to the Action Plan, in which the idea of a CFR was developed for the first time, a major aim of the forthcoming CFR is to serve as a tool for the improvement of the EC law. The future CFR was described in more detail in the Commission's Communication on "European Contract Law and the Revision of the Acquis: The Way Forward" [15]. It proposed that the CFR should provide fundamental principles of contract law, definitions of the main relevant abstract legal terms and model rules of contract law; its main purpose being to serve as a legislators' toolbox.

In drafting the Action Plan the Commission emphasized that the CFR would eliminate market inefficiencies arising from the diverse implementation of European directives, providing a solution to the *non-uniform* interpretation of European contract law due to vague terms and rules, now present in the existing Acquis.

In particular, two issues arise from the vague terminology of EUDs. First, directives adopt broadly defined legal concepts, therefore leaving too much freedom in their implementation to national legislators or judges. Second, directives introduce legal concepts that are different from national legal concepts. Thus, when judges face vague terms, they can either interpret them by referring to the broad principles of the acquis communautaire, or they can refer to the particular goals of the directive in question. To respond to the Action Plan, in the last few years, within the general framework of a "Network of Excellence" European

Project, a research group aiming at consolidating the existing EC law is working on the "Principles of the Existing EC Private Law" or "Acquis Principles" (ACQP). These Principles will be discussed and compared with other outcomes from different European research groups and, during a complex process of consultation with stakeholders under the direction of EC Commission, the CFR will be set up. The Acquis Principles should provide a common terminology as well as common principles to constitute a guideline for uniform implementation and interpretation of European law [5,26]. One outcome of such project is the Acquis Principles glossary, i.e., a set of interconnected terms and concepts.

The Acquis Principles have been sketched by scholars in European Private Law from the so-called *Acquis communautaire*, the existing body of EU primary and secondary legislation as well as the European Court of Justice decisions [5]. Nowadays such corpus contains some 80,000 pages. Notwithstanding the importance of this existing body of settled laws, the Acquis is also a far wider notion, encompassing an impressive set of principles and obligations, going far beyond the internal market and including areas, such as agriculture, environment, energy and transports.

In this paper we show how the multilevel architecture of LTS allows us to relate the Acquis Principles with the legal concepts defined into the directives.

## 3 The Legal Taxonomy Syllabus

In this Section we first summarize the functionalities of the existing LTS [4], and then we explain how it has been extended to cope with the new requirements described in the previous Section.

### 3.1 The basic LTS

The main assumptions of our methodology come from studies in comparative law [24] and ontologies engineering [19]:

- Terms –*lexical entries* for legal information–, and concepts must be distinguished; for this purpose we use lightweight ontologies [17], i.e. simple taxonomic structures of primitive or composite terms together with associated definitions. They are hardly axiomatized as the intended meaning of the terms used by the community is more or less known in advance by all members, and the ontology can be limited to those structural relationships among terms that are considered as relevant [22].
- We distinguish the ontology implicitly defined by EUD, the *EU level*, from the various national ontologies. Each one of these "particular" ontologies belongs to the *national level*: i.e., each national legislation refers to a distinct national legal ontology. We do not assume that the transposition of an EUD automatically introduces in a national ontology the same concepts that are present at the EU level.
- Corresponding concepts at the EU level and at the national level can be denoted by different terms in the same national language.

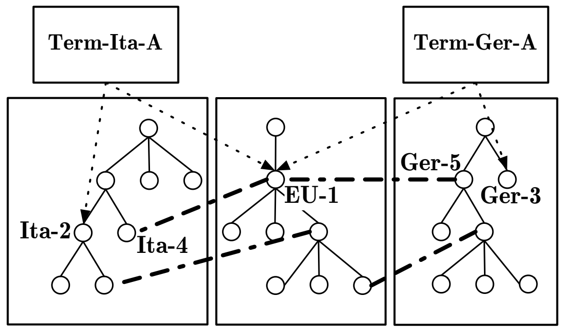


**Fig. 1.** Relationship between ontologies and terms. The thick arcs represent the inter-ontology "association" link.

A standard way to properly manage large multilingual lexical databases is to make a clear distinction among terms and their interlingual acceptions (or *axies*) [27,21].

In the LTS project to properly manage terminological and conceptual misalignment, we distinguish the notion of *legal term* from the notion of *legal concept* and we build a systematic classification based on this distinction. The basic idea in our system is that the conceptual backbone consists in a taxonomy of concepts (ontology) to which the terms can refer in order to express their meaning. One of the main points to keep in mind is that we do not assume the existence of a single taxonomy covering all languages. In fact, the different national systems may organize the concepts in different ways. For instance, the term *contract* corresponds to different concepts in common law and civil law, where it has the meaning of *bargain* and *agreement*, respectively [25]. In most complex instances, there are no homologous between terms-concepts such as *frutto civile* (legal fruit) and *income*, but respectively civil law and common law systems can achieve functionally similar operational rules thanks to the functioning of the entire taxonomy of national legal concepts [18]. Consequently, the LTS includes different ontologies, one for each involved national language plus one for the language of EU documents. Each language-specific ontology is related via a set of *association* links to the EU concepts, as shown in Fig. 1.

Although this picture is conform to intuition, in the basic LTS it has been implemented by taking two issues into account. First, it must be observed that the various national ontologies have a reference language. This is not the case for the EU ontology. For instance, a given term in English could refer either to a concept in the UK ontology or to a concept in the EU ontology. In the first case, the term is used for referring to a concept in the national UK legal system, whilst in the second one, it is used to refer to a concept used in the European directives. This is one of the main advantages of LTS. For example *klar und verständlich* could refer both to concept `Ger-379` (a concept in the German Ontology) and to concept `EU-882` (a concept in the European ontology). This is

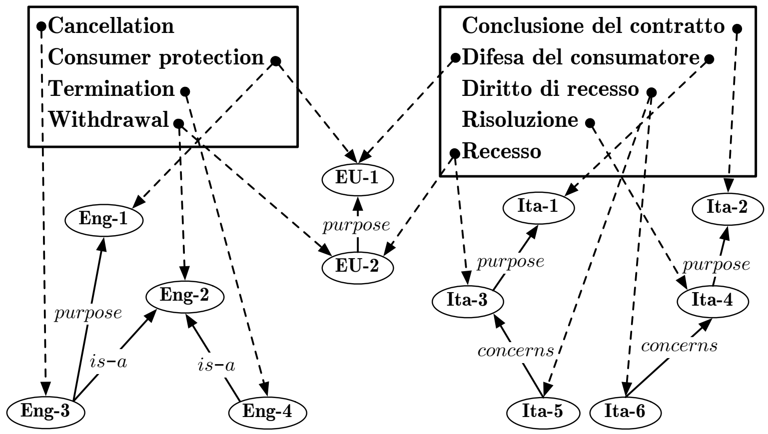


**Fig. 2.** An example of interconnections among terms.

the LTS solution for facing the possibility of a partial correspondence between the meaning of a term in the national system and the meaning of the same term in the translation of a EU directive. This feature enables the LTS to be more precise about what "translation" means. It makes available a way for asserting that two terms are the translation of each other, but just in case those terms have been used in the translation of an EU directive: within LTS, we can talk about direct EU-to-national translations of terms, and about *implicit* national-to-national translations of terms. In other words, we distinguish between *explicit* and *implicit* associations among concepts belonging to different levels. The former ones are direct links that are explicitly used by legal experts to mark a relation between concepts. The latter ones are indirect links: if we start from a concept at a given national level, by following a direct link we reach another concept at European level. Then, we will be able to see how that concept is mapped onto further concepts at the various national levels.

The situation enforced in LTS is depicted in Fig. 1, where it is represented that the Italian term *Term-Ita-A* and the German term *Term-Ger-A* have been used as corresponding terms in the translation of an EU directive, as shown by the fact that both of them refer to the same EU-concept `EU-1`. In the Italian legal system, *Term-Ita-A* has the meaning `Ita-2`. In the German legal system, *Term-Ger-A* has the meaning `Ger-3`. The EU translations of the directive is correct insofar no terms exist in Italian and German that characterize precisely the concept `EU-1` in the two languages (i.e., the "associated" concepts `Ita-4` and `Ger-5` have no corresponding legal terms). A practical example of such a situation is reported in Fig. 2, where we can see that the ontologies include different types of arcs. Beyond the usual *is-a* (linking a category to its supercategory), there are also the arcs *purpose*, which relate a concept to the legal principle motivating it, and *concerns*, which refer to a general relatedness. The dotted arcs represent the reference from terms to concepts. Some terms have links both to a National ontology and to the EU Ontology (in particular, *withdrawal* vs. *recesso* and *difesa del consumatore* vs. *consumer protection*).

The last item above is especially relevant: note that this configuration of arcs specifies that: 1) *withdrawal* and *recesso* have been used as equivalent terms (concept `EU-2`) in some European Directives (e.g., Directive 90/314/EEC). 2) In that context, the term involved an act having as purpose some kind of protection of the consumer. 3) The terms used for referring to the latter are *consumer protection* in English and *difesa del consumatore* in Italian. 4) In the British legal system, however, not all *withdrawals* have this goal, but only a subtype of them, to which the code refers to as *cancellation* (concept `Eng-3`). 5) In the Italian legal system, the term *diritto di recesso* is ambiguous, since it can be used with reference either to something concerning the *risoluzione* (concept `Ita-4`), or to something concerning the *recesso* proper (concept `Ita-3`).

The LTS is a theoretical instrument as well as a software platform that is at the present time yet working. The actual number of annotated terms and concepts are provided in Tables 1 and 2, respectively. Terms were initially extracted from a *corpus* of 24 EC directives, and 2 EC regulations, reported in Appendix 4. Occurrences of such entries were detected from national transposition laws of English, French, Spanish, Italian and German jurisdictions.

**Table 1.** Number of terms

| **Language** | **National** | **European** |
|---|---:|---:|
| French | 8 | 47 |
| Italian | 28 | 52 |
| English | 71 | 75 |
| Spanish | 41 | 60 |
| German | 66 | 98 |
| **total** | 214 | 332 |

**Table 2.** Number of concepts

| **Language** | **National** | **European** |
|---|---:|---:|
| French | 7 | 43 |
| Italian | 24 | 45 |
| English | 54 | 71 |
| Spanish | 34 | 56 |
| German | 52 | 75 |
| **total** | 171 | 290 |

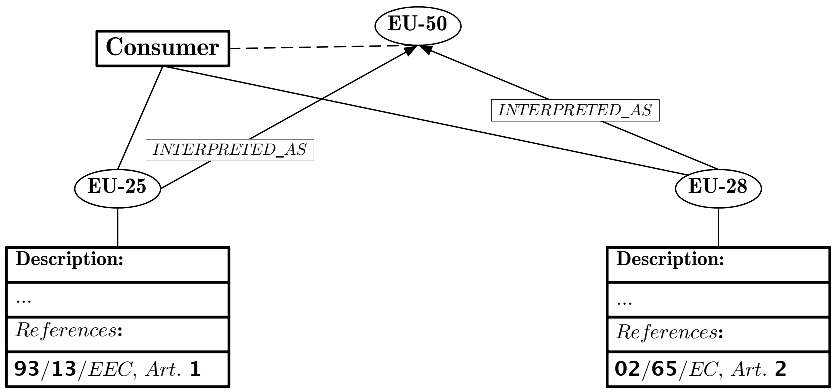


**Fig. 3.** Umeanings `Eu-25` and `Eu-28` are interpreted by the more abstract umeaning `Eu-50`, the link between `Eu-50` and the term "consumer" is implicit.

Finally, it is possible to use the LTS to translate terms into different national systems via the transposed concepts at the European level, i.e. by using the implicit associations. For instance suppose that we want to translate the legal term *credito al consumo* from Italian to German. In the LTS *credito al consumo* is associated to the national umeaning `Ita-175`. We find that `Ita-175` is the transposition of the European umeaning `EU-26` (*contratto di credito*). `EU-26` is associated to the German legal term *Kreditvertrag* at European level. Again, we find that the national German transposition of `EU-26` corresponds to the national umeaning `Ger-32` that is associated with the national legal term *Darlehensvertrag*. Then, by using implicit links in the European ontology, we can translate the Italian legal term *credito al consumo* into the German legal term *Darlehensvertrag*.

### 3.2 Enhancing LTS with interpretation and abstraction

As described in Section 2.2, different pieces of legislations can bear different definitions of terms. Having different detailed definitions is important during the interpretation of very sectorial legal cases, but for the general case it is important to have a view that abstracts from the peculiarities of specific domains.

In order to solve this problem we introduced a new kind of ontologic relation called *INTERPRETED_AS*: it is a non transitive relation where the more general umeaning, that we will call *group leader* represents the abstracted concept that groups the meaning of a number of more specific umeanings, that are the sectorial umeanings defined in the individual EUDs or national laws (see Fig. 3).

We have also introduced a number of constraints and integrity checks to ensure that the semantics of the grouping concept is respected and to improve the usability of the system:

- Each umeaning can belong to a single group.

– A group leader cannot exist without group members.
– When the user searches into the umeaning database, more specific umeanings are excluded from the results unless the user explicitly asks to show them, i.e. only the group leaders are shown in the results.

The need to contextualize concepts to the EUDs defining them leads to the need of more complex instruments to deal with the language of the norms. An umeaning is defined by the legal texts themselves; this makes clear that the creation of umeaning is a quite long task, because it requires from the user searching and reading a very large number of documents.

In order to ease this process, we developed a database that contains the full versions of the desired EUDs and national laws. In this way, the user can carry out his task according to the following workflow.

– The user creates a new *umeaning* linked with the term he wants to define.
– He selects relevant citation from legal text; consequently, the browser is redirected to a search page and the main term attached to the umeaning is used as the default query.
– After choosing one of the search results, the full text of the legal document is displayed, with the search terms highlighted.
– Finally, the user selects the text that will go in the citation with the mouse and confirms the insertion in the references database.

Lastly, when the user searches for a term in the documents database, the search is not performed upon the exact words, rather with their roots, so for instance when performing the search on the term "contracts" also documents containing only "contract" will be found, this seems to enhance the information retrieval performance as shown in [20].

### 3.3 LTS with normative change

When a new normative is approved and enacted it can define a number of new umeanings; moreover it can happen that the same law can change a number of old umeanings defined by old laws. In particular, these old umeaning can become obsolete and no longer valid. We are aware of the difficulties concerning the modelling of the time in artificial intelligence and in formal ontology too[5]. Anyway, in LTS we adopted a naive solution in order to manage the simpler situation concerning t In the LTS it was necessary to delete all old umeanings, causing the loss of all historic informations from the database, informations that are quite valuable to better understand the evolving of the normative. This problem was resolved by using the same solution adopted for the interpretation and abstraction of the norms (Section 3.2), i.e. empowering LTS with a new ontological relation called *REPLACED_BY*.

When the paragraph of an EUD defining an umeaning has been modified by a new EUD, the new one defines a new umeaning that will replace the old

[5] E.g. see [6] for a general survey and [23] for normative systems

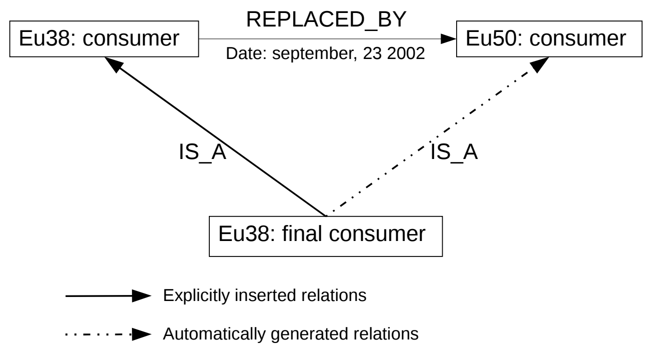


**Fig. 4.** An example of use of the *REPLACED_BY* relation.

umeaning in the ontology. There will be a relation of type *REPLACED_BY* between the two umeanings, where the child umeaning is replaced by the more general umeaning. Also in this case the new ontologic relation has some peculiar characteristics that distinguishes it from the usual ontologic relations (Fig. 4):

- A *REPLACED_BY* relation brings with it a new data field not present in the other relations: the substitution date.
- When the user performs a search in the umeanings database the replaced ones will not be shown, unless the user asks for a certain past date, thus obtaining a snapshot of the legal ontology that was valid in that particular moment.
- When a new umeaning replaces an old one all the ontologic relations where the old umeaning appeared are automatically copied in the new umeaning. If some of them are no longer valid with the new umeaning, manual intervention from the user is required.

### 3.4 Representing a new perspective in LTS: the Acquis Principles Level

One major feature of the LTS approach relies on distinguishing legal information as belonging to different levels. At the current stage of development, the system manages terms and meanings at both EU and national levels. The former one is an ontology of legal concepts derived from the EUDs; the latter one includes national legal ontologies coming from the various national legal systems. The current approach has been devised to be general enough to account for heterogeneous legal sources (like, e.g., EUDs and “Decreti Legislativi” for European and Italian national levels respectively), and flexible enough to be extended by adding further levels. To add a new level into the system, we connect a new legal ontology to an exiting one. The new level is linked via *explicit* associations connecting a concept belonging to the new ontology and a concept belonging to

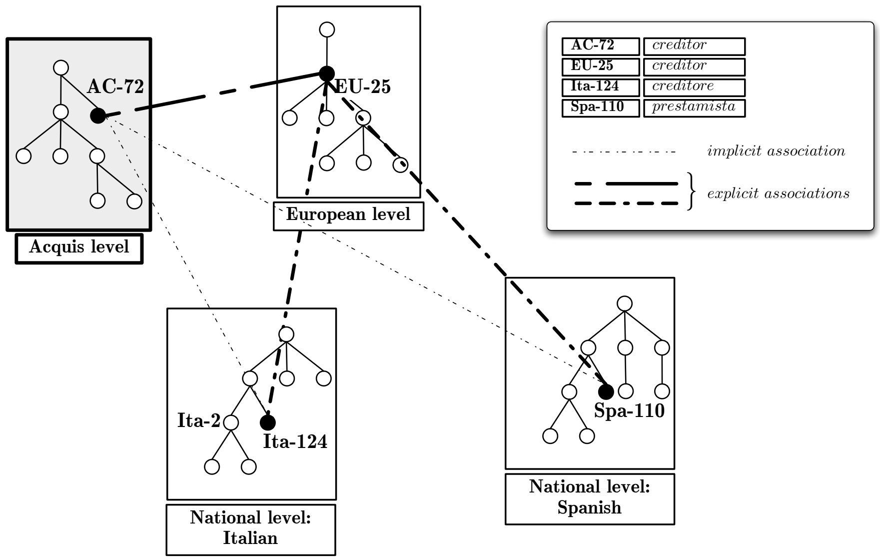


**Fig. 5.** LTS augmented with the Acquis level. Thick lines indicate *explicit* associations; thin lines indicate *implicit* associations.

the existing level ontology. We are applying this procedure in order to define an *Acquis level* to the LTS.

We introduce the *Acquis level* into the LTS by defining explicit associations between Acquis Principles concepts and EU-level concepts. For example, in Fig. 5 we have that the concept `EU-25` (corresponding to the English legal term *creditor*) present in a EUD is explicitly associated to the national legal concepts `Ita-124` (*finanziatore*) and `Spa-110` (*prestamista*) for Italian and Spanish, respectively. We can add the term *creditor* from the Acquis Level by inserting an explicit association between the Acquis legal concept `AC-72` and the European legal concept `EU-25`. As a consequence, the concept `AC-72` is implicitly associated to the legal concepts `Ita-124` and `Spa-110`. This fact has deep consequences on the way one can build systems for reasoning, that are allowed to make paths passing through more than two levels, thereby offering new insights (and ready-to-use associations between terms) to scholars in comparative law.

## 4 Conclusions

In this paper we discuss some new features that have recently been introduced in the LTS, a tool for building multilingual conceptual dictionaries for the EU law. The tool is based on lightweight ontologies to emphasize the distinction between concepts and terms. Different ontologies are built at the EU level and for each national language, to deal with polysemy and terminological and conceptual misalignment.

The present work illustrates how to distinguish between concepts as they are defined in the text of the directives and the concepts representing the doctrinal interpretation of the terms. Moreover, we point out how to deal with normative change by introducing a temporal dimension in ontologies. Finally we show how to add further levels of representation to the EU and national levels. In particular, we described a methodology enabling to insert into the LTS platform a novel set of principles, called Acquis principles. The Acquis principles can be encoded into a number of legal concepts (along with a terminology), and belong to a new level of representation, called Acquis priciples level. These concepts can be integrated into the LTS by new explicit (and implicit) associations, that connect the Acquis principles level with European and national levels.

Two main problems arise in our approach: the first one is theoretical, and it concerns the issue of evaluating the performance of system with more massive data. We would like to show with some quantitative measure the theoretical adequacy of LTS. Secondly, the amount of work needed to annotate the EUDs with concepts, terms and their transpositions, is huge. Future work will involve exploring ways to extend the LTS ontology and populating it at the various levels by semi-automatic approaches [12].

## References


1. G. Ajani, G. Boella, L. Lesmo, M. Martin, A. Mazzei, and P. Rossi. A development tool for multilingual ontology-based conceptual dictionaries. In *Proc. of 5th International Conference on Language Resources and Evaluation, LREC06*, pages 1–6, Genoa, 2006.
2. G. Ajani, G. Boella, L. Lesmo, A. Mazzei, and P. Rossi. Multilingual conceptual dictionaries based on ontologies. In *Proc. of V Legislative XML Workshop*, pages 1–14, Florence, 2006. European Press Academic Publishing.
3. G. Ajani, G. Boella, L. Lesmo, A. Mazzei, and P. Rossi. Multilingual Ontological Analysis of European Directives. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 21–24, Prague, Czech Republic, 2007. Association for Computational Linguistics.
4. G. Ajani, G. Boella, L. Lesmo, A. Mazzei, and P. Rossi. Terminological and ontological analysis of european directives: multilinguism in law. In *11th Internation Conference o Arificial Intelligence and Law (ICAIL)*, pages 43–48, 2007.
5. G. Ajani and H. Schulte-Nölke. The Action Plan on a More Coherent European Contract Law: Response on Behalf of the Acquis Group, 2003.
6. J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.
7. A. Boer, T.M. van Engers, and R. Winkels. Using ontologies for comparing and harmonizing legislation. In *ICAIL*, pages 60–69, 2003.
8. R. Brighi, L. Lesmo, A. Mazzei, M. Palmirani, and D.P. Radicioni. Towards Semantic Interpretation of Legal Modifications through Deep Syntactic Analysis. In *Jurix 2008: The 21st Annual Conference*, Frontiers in Artificial Intelligence and Applications. IOS Press, 2008.

9. M. Cadoli and F. M. Donini. A Survey on Knowledge Compilation. *AI Communications*, 10(3–4):137–150, 1997.
10. P. Casanovas, N. Casellas, C. Tempich, D. Vrandecic, and R. Benjamins. OPJK modeling methodology. In *Proceedings of the ICAIL Workshop: LOAIT 2005*, 2005.
11. M. Cherubini, G. Giardiello, S. Marchi, S. Montemagni, P. Spinosa, and G. Venturi. NLP-based metadata annotation of textual amendments. In *Proc. of WORKSHOP ON LEGISLATIVE XML 2008, JURIX 2008*, 2008.
12. P. Cimiano. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer, 2006.
13. S. Després and S. Szulman. Merging of legal micro-ontologies from european directives. *Journal of Artificial Intelligence and Law*, February 2007.
14. European Commission. Communication from the Commission to the European Parliament and the Council - A More Coherent European Contract Law - An Action Plan. COM, 2003.
15. European Commission. European Contract Law and the revision of the acquis: the way forward. *Communication from the Commission*, COM(2004)(651 final), 2004.
16. E. Giguet and P.S. Luquet. Multilingual lexical database generation from parallel texts in 20 european languages with endogenous resources. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 271–278, Sydney, Australia, 2006. Association for Computational Linguistics.
17. F.o Giunchiglia and I. Zaihrayeu. Lightweight Ontologies. Technical Report DIT-07-071, University of Trento, Department of Information and Communication Technology, 38050 Povo – Trento (Italy), Via Sommarive 14, 2007.
18. M. Graziadei. Tuttifrutti. In P. Birks and A. Pretto, editors, *Themes in Comparative Law*, pages –. Oxford University Press, 2004.
19. M. Klein. Combining and relating ontologies: an analysis of problems and solutions. In *Workshop on Ontologies and Information Sharing, IJCAI'01*, Seattle, USA, 2001.
20. R. J. Krovetz. *Word sense disambiguation for large text databases*. PhD thesis, University of Massachusetts, 1995.
21. V. Lyding, Elena Chiocchetti, G. Sérasset, and F. Brunet-Manquat. The LexALP information system: Term bank and corpus for multilingual legal terminology consolidated. In *Proc. of the Wokshop on Multilingual Language Resources and Interoperability, ACL06*, pages 25–31, 2006.
22. D. Oberle, editor. *Semantic Management of Middleware*. Springer Science+Business and Media, 2005.
23. M. Palmirani and R. Brighi. Time model for managing the dynamic of normative system. *Electronic Government*, pages 207–218, 2006.
24. P. Rossi and C. Vogel. Terms and concepts; towards a syllabus for european private law. *European Review of Private Law (ERPL)*, 12(2):293–300, 2004.
25. R. Sacco. Contract. *European Review of Private Law*, 2:237–240, 1999.
26. R. Schulze. European Private Law and Existing EC Law. *European Review of Private Law (ERPL)*, 2005.
27. G. Sérasset. Interlingual lexical organization for multilingual lexical databases in NADIA. In *Proc. COLING94*, pages 278–282, 1994.
28. The Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. *Nature Genetics* `http://genetics.nature.com`, 25:25–29, 2000.
29. Daniela Tiscornia. The Lois Project: Lexical Ontologies for Legal Information . In *Proceedings of the V Legislative XML Workshop*. European Press Academic Publishing, 2007.

30. P. Vossen, W. Peters, and J. Gonzalo. Towards a universal index of meaning. In *Proc. ACL-99 Siglex Workshop*, 1999.

## APPENDIX A: List of EC directives

### Core directives

- 84/450/EEC concerning misleading advertising
- 85/374/EEC concerning liability for defective products
- 85/577/EEC to protect the consumer in respect of contracts negotiated away from business premises
- 87/102/EEC concerning consumer credit
- 90/88 concerning consumer credit
- 90/314/EEC on package travel, package holidays and package tours
- 93/13/EEC on unfair terms in consumer contracts
- 94/47/EC on the protection of purchasers in respect of certain aspects of contracts relating to the purchase of the right to use immovable properties on a timeshare basis
- 97/7/EC on the protection of consumers in respect of distance contracts
- 97/55/EC concerning misleading advertising so as to include comparative advertising
- 98/6 on consumer protection in the indication of the prices of products offered to consumers
- 98/7 concerning consumer credit
- 98/27/EC on injunctions for the protection of consumers' interests
- 99/44/EC on certain aspects of the sale of consumer goods and associated guarantees
- 2000/13/EC relating to labelling, presentation and advertising of foodstuff
- 2001/95 on general product safety
- 2002/65/EC concerning the distance marketing of consumer financial services
- Regulation 2006/2004/EC on co-operation between national authorities responsible for the enforcement of consumer protection laws
- Directive 2005/29/EC on Unfair Commercial Practices

### Ancillary directives

- 76/768/EEC relating to cosmetic products
- 88/378/EEC on toy safety
- 89/552/EEC on TV broadcasting activities
- 96/74/EC on textile names
- 97/5/EC on cross border credit transfers
- Recommendation 98/257 on the principles applicable to bodies responsible for the out-of-court settlement of consumer disputes
- 2000/31/EC on electronic commerce
- Regulation 2560/2001/EC on cross-border payments in Euro

# Prototypical reasoning with low complexity Description Logics: preliminary results

Laura Giordano*, Valentina Gliozzi⊙, Nicola Olivetti*, Gian Luca Pozzato⊙

* Dip. di Informatica - Univ. Piemonte O. "A. Avogadro" - laura@mfn.unipmn.it
⊙ Dip. di Informatica - Università di Torino - {gliozzi,pozzato}@di.unito.it
* LSIS-UMR CNRS 6168 Univ. "P. Cézanne" - nicola.olivetti@univ-cezanne.fr

**Abstract.** We present an extension $\mathcal{EL}^{+^\perp}\mathbf{T}$ of the description logic $\mathcal{EL}^{+^\perp}$ for reasoning about prototypical properties and inheritance with exceptions. $\mathcal{EL}^{+^\perp}\mathbf{T}$ is obtained by adding to $\mathcal{EL}^{+^\perp}$ a typicality operator $\mathbf{T}$, which is intended to select the "typical" instances of a concept. In $\mathcal{EL}^{+^\perp}\mathbf{T}$ knowledge bases may contain inclusions of the form "$\mathbf{T}(C)$ is subsumed by $P$", expressing that typical $C$-members have the property $P$. We show that the problem of entailment in $\mathcal{EL}^{+^\perp}\mathbf{T}$ is in co-NP.

## 1 Introduction

In Description Logics (DLs) the need of representing prototypical properties and of reasoning about defeasible inheritance of such properties naturally arises. The traditional approach is to handle defeasible inheritance by integrating some kind of nonmonotonic reasoning mechanism. This has led to study nonmonotonic extensions of DLs [2–6, 12]. However, finding a suitable nonmonotonic extension for inheritance with exceptions is far from obvious.

In this work we introduce a defeasible extension of the description logic $\mathcal{EL}^{+^\perp}$ called $\mathcal{EL}^{+^\perp}\mathbf{T}$, continuing the investigation started in [7], where we extended the logic $\mathcal{ALC}$ with a typicality operator $\mathbf{T}$. The intended meaning of the operator $\mathbf{T}$ is that, for any concept $C$, $\mathbf{T}(C)$ singles out the instances of $C$ that are considered as "typical" or "normal". Thus assertions as "typical football players love football" are represented by $\mathbf{T}(\mathit{FootballPlayer}) \sqsubseteq \mathit{FootballLover}$. The semantics of the typicality operator $\mathbf{T}$ turns out to be strongly related to the semantics of nonmonotonic entailment in KLM logic $\mathbf{P}$ [11].

In our setting, we assume that the TBox element of a KB comprises, in addition to the standard concept inclusions, a set of inclusions of the type $\mathbf{T}(C) \sqsubseteq D$ where $D$ is a concept not mentioning $\mathbf{T}$. For instance, a KB may contain: $\mathbf{T}(\mathit{Dog}) \sqsubseteq \mathit{Affectionate}$; $\mathbf{T}(\mathit{Dog}) \sqsubseteq \mathit{CarriedByTrain}$; $\mathbf{T}(\mathit{Dog} \sqcap \mathit{PitBull}) \sqsubseteq \mathit{NotCarriedByTrain}$; $\mathit{CarriedByTrain} \sqcap \mathit{NotCarriedByTrain} \sqsubseteq \bot$, corresponding to the assertions: typically dogs are affectionate, normally dogs can be transported by train, whereas typically a dog belonging to the race of pitbull cannot (since pitbulls are considered as reactive dogs); the fourth inclusion represents the disjointness of the two concepts $\mathit{CarriedByTrain}$ and $\mathit{NotCarriedByTrain}$. Notice that, in standard DLs, replacing the second and the third inclusion with $\mathit{Dog} \sqsubseteq \mathit{CarriedByTrain}$ and $\mathit{Dog} \sqcap \mathit{PitBull} \sqsubseteq \mathit{NotCarriedByTrain}$, respectively, we would simply get that there are not pitbull dogs, thus the KB would collapse.

This collapse is avoided as we do not assume that $\mathbf{T}$ is monotonic, that is to say $C \sqsubseteq D$ does not imply $\mathbf{T}(C) \sqsubseteq \mathbf{T}(D)$.

By the properties of $\mathbf{T}$, some inclusions are entailed by the above KB, as for instance $\mathbf{T}(Dog \sqcap CarriedByTrain) \sqsubseteq Affectionate$. In our setting we can also use the $\mathbf{T}$ operator to state that some domain elements are typical instances of a given concept. For instance, an ABox may contain either $\mathbf{T}(Dog)(fido)$ or $\mathbf{T}(Dog \sqcap PitBull)(fido)$. In the two cases, the expected conclusions are entailed: $CarriedByTrain(fido)$ and $NotCarriedByTrain(fido)$, respectively.

In this work, we present some preliminary results on *low complexity* Description Logics extended with the typicality operator $\mathbf{T}$. In particular we focus on the logic $\mathcal{EL}^{+^\perp}$ of the well known $\mathcal{EL}$ family. The logics of the $\mathcal{EL}$ family allow for conjunction ($\sqcap$) and existential restriction ($\exists R.C$). Despite their relatively low expressivity, a renewed interest has recently emerged for these logics. Indeed, theoretical results have shown that $\mathcal{EL}$ has better algorithmic properties than its counterpart $\mathcal{FL}_0$, which allows for conjunction and value restriction ($\forall R.C$). Also, it has turned out that the logics of the $\mathcal{EL}$ family are relevant for several applications, in particular in the bio-medical domain; for instance, medical terminologies, such as GALEN, SNOMED, and the Gene Ontology used in bioinformatics, can be formalized in small extensions of $\mathcal{EL}$.

We present some results about the complexity of $\mathcal{EL}^{+^\perp}\mathbf{T}$. We show that, given an $\mathcal{EL}^{+^\perp}\mathbf{T}$ KB, if it is satisfible, then there is a *small* model whose size is polynomial in the size of KB. The construction of the model exploits the facts that (1) it is possible to reuse the same domain element (instance of a concept $C$) to fulfill existential formulas $\exists r.C$ w.r.t. domain elements; (2) we can restrict our attention to a class of models in which the preference relation $<$ is multilinear and polynomial, that is it determines a set of disjoint chains of elements of polynomial length. The construction of the model allows us to conclude that the problem of deciding entailment in $\mathcal{EL}^{+^\perp}\mathbf{T}$ is in co-NP.

Technical details and proofs can be found in the accompanying report [10].

## 2 The Logic $\mathcal{EL}^{+^\perp}\mathbf{T}$

We consider an alphabet of concept names $\mathcal{C}$, of role names $\mathcal{R}$, and of individuals $\mathcal{O}$. The language $\mathcal{L}$ of the logic $\mathcal{EL}^{+^\perp}\mathbf{T}$ is defined by distinguishing *concepts* and *extended concepts* as follows: (Concepts) $A \in \mathcal{C}$, $\top$, and $\perp$ are *concepts* of $\mathcal{L}$; if $C, D \in \mathcal{L}$ and $r \in \mathcal{R}$, then $C \sqcap D$ and $\exists r.C$ are *concepts* of $\mathcal{L}$. (Extended concepts) if $C$ is a concept, then $C$ and $\mathbf{T}(C)$ are extended concepts of $\mathcal{L}$. A knowledge base is a pair (TBox,ABox). TBox contains (i) a finite set of GCIs $C \sqsubseteq D$, where $C$ is an extended concept (either $C'$ or $\mathbf{T}(C')$), and $D$ is a concept, and (ii) a finite set of role inclusions (RIs) $r_1 \circ r_2 \circ \cdots \circ r_n \sqsubseteq r$. ABox contains expressions of the form $C(a)$ and $r(a,b)$ where $C$ is an extended concept, $r \in \mathcal{R}$, and $a, b \in \mathcal{O}$. In order to provide a semantics to the operator $\mathbf{T}$, we extend the definition of a model used in "standard" terminological logic $\mathcal{EL}^{+^\perp}$:

**Definition 1 (Semantics of T).** *A model $\mathcal{M}$ is any structure $\langle \Delta, <, I \rangle$, where $\Delta$ is the domain; $<$ is an irreflexive and transitive relation over $\Delta$, and satisfies* *the following* Smoothness Condition*: for all $S \subseteq \Delta$, for all $a \in S$, either $a \in Min_<(S)$ or $\exists b \in Min_<(S)$ such that $b < a$, where $Min_<(S) = \{a : a \in S$ and $\nexists b \in S$ s.t. $b < a\}$. $I$ is the extension function that maps each extended concept $C$ to $C^I \subseteq \Delta$, and each role $r$ to a $r^I \subseteq \Delta^I \times \Delta^I$. For concepts of $\mathcal{EL}^{+^\perp}$, $C^I$ is defined in the usual way. For the $\mathbf{T}$ operator: $(\mathbf{T}(C))^I = Min_<(C^I)$. A model satisfying a KB (TBox,ABox) is defined as usual. Moreover, we assume the unique name assumption.*

Notice that the meaning of $\mathbf{T}$ can be split into two parts: for any $a$ of the domain $\Delta$, $a \in (\mathbf{T}(C))^I$ just in case (i) $a \in C^I$, and (ii) there is no $b \in C^I$ such that $b < a$. In order to isolate the second part of the meaning of $\mathbf{T}$, we introduce a new modality $\Box$. The basic idea is simply to interpret the preference relation $<$ as an accessibility relation. By the Smoothness Condition, it turns out that $\Box$ has the properties as in Gödel-Löb modal logic of provability G. The interpretation of $\Box$ in $\mathcal{M}$ is as follows: $(\Box C)^I = \{a \in \Delta \mid$ for every $b \in \Delta$, if $b < a$ then $b \in C^I\}$. We have that $a$ is a typical instance of $C$ ($a \in (\mathbf{T}(C))^I$) iff $a \in C^I$ and, for all $b < a$, $b \notin C^I$, namely we have that $a \in (\mathbf{T}(C))^I$ iff $a \in (C \sqcap \Box\neg C)^I$. From now on, we consider $\mathbf{T}(C)$ as an abbreviation for $C \sqcap \Box\neg C$. The Smoothness Condition ensures that typical elements of $C^I$ exist whenever $C^I \neq \emptyset$, by preventing infinitely descending chains of elements.

## 3 Complexity of $\mathcal{EL}^{+^\perp}\mathbf{T}$

In order to give a complexity upper bound for the logic $\mathcal{EL}^{+^\perp}\mathbf{T}$, we show that, given a model $\mathcal{M} = \langle \Delta, <, I\rangle$ of a KB, we can build a *small* model of KB whose size is polynomial in the size of the KB.

**Theorem 1 (Small model theorem).** *Let KB=(TBox,ABox) be an $\mathcal{EL}^{+^\perp}\mathbf{T}$ knowledge base. For all models $\mathcal{M} = \langle \Delta, <, I\rangle$ of KB and all $x \in \Delta$, there exists a model $\mathcal{N} = \langle \Delta^\circ, <^\circ, I^\circ\rangle$ of KB such that (i) $x \in \Delta^\circ$, (ii) for all $\mathcal{EL}^{+^\perp}\mathbf{T}$ concepts $C$, $x \in C^I$ iff $x \in C^{I^\circ}$, and (iii) $\mid \Delta^\circ \mid$ is polynomial in the size of KB.*

Due to space limitations, here we only give a sketch of the proof, whose details can be found in [10]. The construction comprises three steps.

(step A) First of all, in order to reduce the size of the model, we cut a portion of it that includes $x$. We build a model $\mathcal{M}'$ by means of the following construction. For each atomic concept $C \in \mathcal{C}$ and for each role $r \in \mathcal{R}$ we let $S(C)$ and $R(r)$ be the mappings computed by the algorithm defined in [1] to compute subsumption by means of completion rules. As usual, for a given individual $a$ in the ABox, we write $a^I$ to denote the element of $\Delta$ corresponding to the extension of $a$ in $\mathcal{M}$. We make use of three sets of elements: $\Delta_0$ will be part of the domain of the model being constructed, and it contains a portion of the domain $\Delta$ of the initial model. All elements introduced in the domain must be processed in order to satisfy the existential formulas. *Unres* is used to keep track of not yet processed elements. Finally, $\Delta_1$ is a set of elements that will belong to the domain of the constructed model. Each element $w_C$ of $\Delta_1$ is created for a corresponding atomic concept $C$ and is used to satisfy any existential formula $\exists r.C$ throughout the model. In the following by $w_C$ we mean the domain element of $\Delta_1$ which is added for the atomic concept $C$. We provide an algorithmic description of the construction of model $\mathcal{M}'$ from the given model $\mathcal{M}$. Observe that $\mathcal{M}$ can be an infinite model.

1. $\Delta_0 := \{x\} \cup \{a^I \in \Delta \mid a$ occurs in the ABox $\}$
2. *Unres*$:=\{x\} \cup \{a^I \in \Delta \mid a$ occurs in the ABox $\}$
3. $\Delta_1 := \emptyset$
4. **while** *Unres* $\neq \emptyset$ **do**
5.     extract one $y$ from *Unres*
6.     **for each** $\exists r.C$ occurring in KB s.t. $y \in (\exists r.C)^I$ **do**
7.         **if** $\nexists w_C \in \Delta_1$ **then**
8.             choose $w \in \Delta$ s.t. $(y, w) \in r^I$ and $w \in C^I$
9.             $\Delta_0 := \Delta_0 \cup \{w\}$
10.             *Unres*:=*Unres* $\cup \{w\}$
11.             create a new element $w_C$ associated with $C$
12.             $\Delta_1 := \Delta_1 \cup \{w_C\}$
13.             add $w <' w_C$
14.             add $(y, w_C)$ to $r^{I'}$
15.         **else**
16.             add $(y, w_C)$ to $r^{I'}$
17.     **for each** $y_i \in \Delta$ such that $y_i < y$ **do**
18.         $\Delta_0 := \Delta_0 \cup \{y_i\}$
19.         *Unres*:=*Unres* $\cup \{y_i\}$
20. **for each** $w_C, w_D \in \Delta_1$ with $C \neq D$ **do**
21.     **if** $(C, D) \in R(r)$ **then** add $(w_C, w_D)$ to $r^{I'}$

The model $\mathcal{M}' = \langle \Delta', <', I' \rangle$ is defined as follows:

- $\Delta' = \Delta_0 \cup \Delta_1$
- we extend $<'$ computed by the algorithm by adding $u <' v$ if $u < v$, for each $u, v \in \Delta'$;
- the extension function $I'$ is defined as follows: • for all atomic concepts $C \in \mathcal{C}$, for all domain elements in $\Delta'$, we define: for each $u \in \Delta_0$, we let $u \in C^{I'}$ if $u \in C^I$; for each $w_D \in \Delta_1$, we let $w_D \in C^{I'}$ if $C \in S(D)$. • for all roles $r$, we extend $r^{I'}$ constructed by the algorithm by means of the following role closure rules: for all inclusions $r \sqsubseteq s \in$ TBox, if $(u, v) \in r^{I'}$ then add $(u, v)$ to $s^{I'}$; for all inclusions $r_1 \circ r_2 \sqsubseteq s \in$ TBox, if $(u, v) \in r_1^{I'}$ and $(v, w) \in r_2^{I'}$ then add $(u, w)$ to $s^{I'}$. • $I'$ is extended so that it assigns $a^I$ to each individual $a$ in the ABox.

$\mathcal{M}'$ is not guaranteed to have polynomial size in the KB because in line 18 we add an element $y_i$ for each $y_i < y$, then the size of $\Delta_0$ may be arbitrarily large.

(step B) We refine our construction in order to obtain from $\mathcal{M}'$ a multi-linear model with a polynomial number of chains. Intuitively, a model is *multi-linear* if the relation $<$ forms a set of chains of domain elements, that is, for every $u, v, z$ of the domain, we have that: (i) if $u < z$ and $v < z$ and $u \neq v$, then $u < v$ or

$v < u$; (ii) if $z < u$ and $z < v$ and $u \neq v$, then $u < v$ or $v < u$. From $\mathcal{M}'$ we can obtain a multilinear model $\mathcal{M}''$ that preserves the interpretation of atomic concepts with respect to common elements of the domain and has a polynomial number of chains.

(step C) We finally construct a model $\mathcal{N}$ from $\mathcal{M}''$ whose domain has polynomial size in the size of KB. The idea is as follows. Let us consider a chain $w_0, w_1, w_2, \ldots$ in the multi-linear model. We can observe that, given $w_i$ and $w_j$ in the chain such that $w_i < w_j$, the set of negated box formulas $\neg\Box\neg C$ of which $w_i$ is an instance is a subset of the set of negated box formulas of which $w_j$ is an instance. We can thus shrink each chain by retaining only the elements $w_i$, $w_j$ such that $w_i < w_j$ implies there exists a formula $\neg\Box\neg C$ such that $w_j$ is an instance of $\neg\Box\neg C$ and $w_i$ is not an instance of $\neg\Box\neg C$. As there is only a polynomial number of such box formulas $\neg\Box\neg C$, each chain will contain only a polynomial number of elements. Since the number of chains is polynomial in itself (by step B), the resulting model $\mathcal{N}$ has a polynomial size.

Given Theorem 1 above, when evaluating the entailment, we can restrict our consideration to small models, namely, to polynomial multi-linear models of the KB. We write KB $\models \alpha$ to say that a query $\alpha$ holds in all the models of the KB. A query $\alpha$ is either a formula of the form $C(a)$ or a subsumption relation $C \sqsubseteq D$. We write KB $\models_s \alpha$ to say that $\alpha$ holds in all polynomial multi-linear models of the KB. It holds that KB $\models \alpha$ if and only if KB $\models_s \alpha$. As a consequence, we can give an upper bound on the complexity of $\mathcal{EL}^{+^{\perp}}\mathbf{T}$:

**Theorem 2.** *In $\mathcal{EL}^{+^{\perp}}\mathbf{T}$, the problem of deciding whether KB $\models \alpha$ is in* co-NP*. The problems of satisfiability of a KB and of concept satisfiability are in* NP*. The problems of subsumption and of instance checking are in* co-NP*.*

## 4 Conclusions and future issues

We have presented the description logic $\mathcal{EL}^{+^{\perp}}\mathbf{T}$, that is $\mathcal{EL}^{+^{\perp}}$ extended by a tipicality operator $\mathbf{T}$ intended to select the "most normal" instances of a concept. Whereas for $\mathcal{ALC} + \mathbf{T}$ deciding satisfiability (subsumption) is EXPTIME complete (see [9]), we have shown here that for $\mathcal{EL}^{+^{\perp}}\mathbf{T}$ the complexity is significantly smaller, namely it reduces to NP for satisfiability (and co-NP for subsumption). This result is obtained by a "small" model property (of a particular kind: multi-linear) that fails for the whole $\mathcal{ALC} + \mathbf{T}$ as well as for $\mathcal{ALC}$. We believe that this bound is also a lower bound, but we have not proved it so far. Although validity/satisfiability for KLM logic $\mathbf{P}$ is known to be (co)NP hard, in $\mathcal{EL}^{+^{\perp}}\mathbf{T}$, we can only directly encode nonmonotonic assertions $A \mid\sim B$ where $A$ is a conjunction of atoms and $B$ is either an atom or $\perp$. As far as we know, the complexity of this fragment of $\mathbf{P}$ is unknown. Thus a lower bound for $\mathcal{EL}^{+^{\perp}}\mathbf{T}$ cannot be obtained from known results about KLM logic $\mathbf{P}$.

The logic $\mathcal{EL}^{+^{\perp}}\mathbf{T}$ in itself is not sufficient for prototypical reasoning and inheritance with exceptions, in particular we need a stronger (nonmonotonic) mechanism to cope with the problem known as *irrelevance*. Concerning the example of the Introduction, we would like to conclude that typical red dogs are affectionate, since the color of a dog is irrelevant with respect to the property of being affectionate. However, as the property of being red is not a property neither of all dogs, nor of typical dogs, in $\mathcal{EL}^{+^{\perp}}\mathbf{T}$ we are not able to conclude $\mathbf{T}(Dog \sqcap Red) \sqsubseteq Affectionate$. One possibility is to consider a stronger (nonmonotonic) entailment relation $\mathcal{EL}^{+^{\perp}}\mathbf{T}_{min}$ determined by restricting the entailment of $\mathcal{EL}^{+^{\perp}}\mathbf{T}$ to "minimal models", as defined in [8] for $\mathcal{ALC} + \mathbf{T}$. Intuitively, minimal models are those that maximise "typical instances" of a concept. As shown in [8], for $\mathcal{ALC} + \mathbf{T}_{min}$, minimal entailment can be decided in co-NExp$^{\text{NP}}$. We believe that for $\mathcal{EL}^{+^{\perp}}\mathbf{T}_{min}$ we can obtain a smaller complexity upper bound on the base of the results presented here.

**Acknowledgements.** The work was partially supported by Regione Piemonte, Project "ICT4Law - *ICT Converging on Law: Next Generation Services for Citizens, Enterprises, Public Administration and Policymakers*".

## References

1. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of IJCAI'05, Professional Book Center* pp. 364-369, 2005.
2. F. Baader and B. Hollunder. Embedding defaults into terminological knowledge representation formalisms. *J. Autom. Reasoning*, 14(1):149–180, 1995.
3. F. Baader and B. Hollunder. Priorities on defaults with prerequisites, and their application in treating specificity in terminological default logic. *J. of Automated Reasoning (JAR)*, 15(1):41–68, 1995.
4. P. A. Bonatti, C. Lutz, and F. Wolter. Description logics with circumscription. In *Proc. of KR*, pages 400–410, 2006.
5. F. M. Donini, D. Nardi, and R. Rosati. Description logics of minimal knowledge and negation as failure. *ACM Trans. Comput. Log.*, 3(2):177–225, 2002.
6. T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining answer set programming with description logics for the semantic web. In *KR 2004, 141-151.*
7. L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. Preferential Description Logics. In LPAR 2007. LNAI 4790, pp. 257-272, 2007.
8. L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. Reasoning About Typicality in Preferential Description Logics. In JELIA 2008. LNAI 5293, pp. 192-205, 2008.
9. L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. On Extending Description Logics for Reasoning About Typicality: a First Step. In *Technical Report 116/09, Dip. di Informatica, Univ. di Torino*, 2009.
10. L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. Reasoning About Typicality in Low Complexity Description Logics: Preliminary Results. In *Technical Report 121/09, Dip. di Informatica, Univ. di Torino*, 2009.
11. S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44(1-2):167–207, 1990.
12. U. Straccia. Default inheritance reasoning in hybrid kl-one-style logics. In *Proc. of IJCAI*, pages 676–681, 1993.

# Verifying Agents Conformance with Multiparty Protocols

Laura Giordano[1] and Alberto Martelli[2]

[1] Dipartimento di Informatica, Università del Piemonte Orientale, Alessandria
[2] Dipartimento di Informatica, Università di Torino, Torino

**Abstract.** The paper deals with the problem of agents conformance with multiparty protocols. We introduce a notion of conformance of a set of k agents with a multiparty protocol with $k$ roles, which requires the agents to be interoperable and to produce correct executions of the protocol. We introduce conditions that enable each agent to be independently verified with respect to the protocol. We assume that protocols are specified in a temporal action theory and we show that the problem of verifying the conformance of an agent with a protocol can be solved by making use of automata based techniques. Protocols with nonterminating computations, modeling reactive agents, can also be captured in this framework.

## 1 Introduction

In an open environment, the interaction of agents is ruled by interaction protocols on which agents commonly agree. An important issue, in this regard, concerns agent conformance with the protocol. Although agent policy may somehow deviate from the behavior dictated by the protocol, in some cases we want, nevertheless, to regard the policy as being compatible with the protocol. A related issue concerns the *interoperability* of agents in an open environment. The need for conditions to guarantee that a set of agents may interact properly, has led to the introduction of different notions of *compatibility* among agents [4] as well as to the definition of notions of *conformance* of an agent with a protocol [2,5,11,17].

In this paper, we define a notion of agent conformance for the general case of multiparty protocols. This notion must assure that a set of agents, that are conformant with a protocol, interoperate (in particular, they do not get stuck) and that their interactions produce correct executions of the protocol.

In our proposal, the specification of agents and protocols is given in a temporal action theory [9,13], by means of temporal constraints, and the communication among agents is synchronous. Protocols are given a *declarative specification* consisting of: (i) the specification of communicative actions by means of their effects and preconditions on the social state which, in particular, includes commitments; (ii) a set of temporal constraints, which specify the wanted interactions (and, under this respect, our approach to protocol specification is similar to the

M. Fisher, F. Sadri, and M. Thielscher (Eds.): CLIMA IX, LNAI 5405, pp. 17–36, 2009.

one proposed in *DecSerFlow* [21]). Protocols with nonterminating computations, modeling reactive services [8], can also be captured in this framework.

We define a multiparty protocol $P$ with $k$ roles, by separately specifying the behavior of all roles $P_1, \ldots, P_k$ in the protocol. We then introduce a notion of *interoperability* among a set of agents, which guarantees the agents to interact properly. More precisely, each agent can freely choose among its possible emissions without the computation getting stuck.

Given a multiparty protocol $P$, we define a notion of *conformance of a set of agents* $A_1, \ldots, A_k$ with $P$: agents $A_1, \ldots, A_k$ interoperate and their interaction only produces runs of the protocol $P$. Verifying the conformance of a set of agents all together, however, is not feasible in an open environment, as, in general, the internal behavior of all agents participating in a protocol is not known. The verification of each agent participating in the protocol must be done independently.

In this paper, we introduce a definition of *conformance of a single agent* $A_i$ *(playing role i) with the protocol* $P$. We will see that verifying an agent $A_i$ with respect with its role $P_i$ is not sufficient to guarantee the interoperability of a set of conformant agents in the multiparty case, unless a rather narrow notion of conformance is adopted. Indeed, our notion of conformance of an agent $A_i$ with a protocol $P$ is defined by comparing the runs of agent $A_i$, when executed in the context of the protocol, and the runs of $P$ itself. We prove that a set of agents which are independently conformant with the protocol $P$ are guaranteed to be interoperable and to produce correct executions of $P$.

Starting from a specification of the protocol in a temporal logic, the problem of verifying the conformance of an agent with a protocol can be solved by making use of an automata based approach. In particular, interoperability can be checked by working on the Büchi automaton which can be extracted from the logical specification of the protocol.

## 2 Protocol Specification

The specification of interaction protocols we adopt is based on Dynamic Linear Time Temporal Logic (DLTL) [15], a linear time temporal logic which extends LTL by allowing the until operator to be indexed by programs in Propositional Dynamic Logic (PDL). DLTL allows until formulas of the form $\alpha \mathcal{U}^{\pi} \beta$, where the program $\pi \in Prg(\Sigma)$ is a regular expression built from a set $\Sigma$ of atomic actions. More precisely, $Prg(\Sigma) ::= a \mid \pi_1 + \pi_2 \mid \pi_1; \pi_2 \mid \pi^*$, where $a \in \Sigma$ and $\pi_1, \pi_2, \pi$ range over $Prg(\Sigma)$.

As for LTL, DLTL models are infinite linear sequences of worlds (propositional interpretations), each one reachable from the initial world by a finite sequence $\tau$ of actions in the alphabet $\Sigma$. More precisely, a model $M = (\sigma, V)$ consists of an infinite sequence of actions $\sigma$ over $\Sigma$ (the sequence of actions executed from the initial world) and a valuation function $V$, defining the interpretation of propositions at each world $\tau$ (where $\tau$ is a prefix of $\sigma$).

In the following, we denote by $prf(\sigma)$ the set of all finite prefixes of $\sigma$ (the worlds) and, for each regular program $\pi$, we denote by $[[\pi]]$ the set of finite sequences associated with $\pi$. Given a model $M = (\sigma, V)$, a finite word $\tau \in prf(\sigma)$ and a formula $\alpha$, the *satisfiability of a formula $\alpha$ at $\tau$ in $M$*, written $M, \tau \models \alpha$, is defined as usual for the classical connectives. Moreover:

- $M, \tau \models p$ iff $p \in V(\tau)$;
- $M, \tau \models \alpha \mathcal{U}^{\pi} \beta$ iff there exists $\tau' \in [[\pi]]$ such that $\tau\tau' \in prf(\sigma)$ and $M, \tau\tau' \models \beta$. Moreover, for every $\tau''$ such that $\varepsilon \leq \tau'' < \tau'$, $M, \tau\tau'' \models \alpha$.

A formula $\alpha \mathcal{U}^{\pi} \beta$ is true at a world $\tau$ if "$\alpha$ until $\beta$" is true on a finite stretch of behavior which is in the linear time behavior of the program $\pi$.

The derived modalities $\langle\pi\rangle$ and $[\pi]$ can be defined as follows: $\langle\pi\rangle\alpha \equiv \top \mathcal{U}^{\pi} \alpha$ and $[\pi]\alpha \equiv \neg\langle\pi\rangle\neg\alpha$. When $\pi$ is $\Sigma^*$ (representing all finite actions sequences), we replace $\langle\pi\rangle$ with $\Diamond$ and $[\pi]$ with $\Box$. Furthermore, the $\bigcirc$ (next) operator can be defined as $\bigcirc\alpha \equiv \bigvee_{a \in \Sigma} \langle a \rangle \alpha$. As shown in [15], DLTL($\Sigma$) is strictly more expressive than LTL($\Sigma$). The satisfiability and validity problems for DLTL are PSPACE complete problems [15].

In this paper, we make use of the Product version of DLTL, $DLTL^{\otimes}$ [14], which allows to describe the behavior of a network of sequential agents which coordinate their activities by performing common actions together. There are $k$ agents $1, \ldots, k$, and a *distributed alphabet* $\tilde{\Sigma} = \{\Sigma_i\}_{i=1}^{k}$, a family of (possibly non-disjoint) alphabets, with each $\Sigma_i$ a non-empty, finite set of actions ($\Sigma_i$ is the set of actions which require the participation of agent $i$).

Atomic propositions are introduced in a local fashion, by introducing a non-empty set of atomic propositions $\mathcal{P}$. For each atomic proposition $p \in \mathcal{P}$ and agent $i$, $p_i$ represents the "local" view of the proposition $p$ at $i$, and is evaluated in the local state of agent $i$. The formulas of the language are obtained as the boolean combination of the formulas of $DLTL_i^{\otimes}$ which can be constructed on the alphabet (actions and propositions) of each agent $i$, using the modalities $\mathcal{U}_i^{\pi}$, $\langle\pi\rangle_i$, $[\pi]_i$, $\bigcirc_i$, $\Diamond_i$ and $\Box_i$. A $DLTL^{\otimes}$ model is a pair $(\sigma, V)$, where $V = \{V_i\}_{i=1}^{k}$ is a family of valuation functions, one for each agent $i$. The satisfiability of the formulas of $DLTL_i^{\otimes}$ is evaluated by making use of $V_i$ and of the projection $\sigma|_i$ of $\sigma$ to $\Sigma_i$ (where $\sigma|_i$ is the sequence obtained by erasing from $\sigma$ all occurrences of symbols that are not in $\Sigma_i$).

We illustrate how a protocol can be specified in this framework trough the specification of a *Purchase protocol*.

*Example 1.* We have three roles: the *merchant (mr)*, the *customer (ct)* and the *bank (bk)*. *ct* sends a request to *mr*; *mr* replies with an offer or by saying that the requested good is not available. If *ct* receives the offer, it may either accept the offer and send a payment request to *bk*, or refuse the offer. If *ct* accepts the offer, then *mr* delivers the goods. If *ct* requires *bk* to pay *mr*, *bk* sends the payment. *ct* can send the request for payment to *bk* even before it has received the goods.

In this example, all actions are communicative actions:*sendRequest, sendOffer, sendNotAvail, sendAccept, sendRefuse, sendPaymentReqest, sendPayment*, each one belonging to the action alphabet of the sender and of the receiver. For instance, action *sendRequest* is both in $\Sigma_{ct}$ (as *ct* is the sender of the request) and in $\Sigma_{mr}$ (as *mr* is the receiver of the request). Communication is synchronous: roles communicate by synchronizing on the execution of communicative actions.

The Purchase protocol $Pu$ is given by specifying separately the protocols of the three participating roles: $P_{ct}, P_{mr}$ and $P_{bk}$. The role $P_i$ in the protoocol is specified by a *domain description* $D_i$, which is a pair $(\Pi_i, \mathcal{C}_i)$, where $\Pi_i$ is a set of formulas describing the effects and preconditions of the actions (the action theory) of role $i$, and $\mathcal{C}_i$ is a set of *constraints* that the executions of role $i$ must satisfy. The approach is a generalization of the one proposed in [13].

Let us define, for instance, the *domain description* $D_{mr} = (\Pi_{mr}, \mathcal{C}_{mr})$ of the merchant. We adopt a social approach where an interaction protocol is specified by describing the effects of communicative actions on the social state. The social state contains the domain specific fluents describing observable facts concerning the execution of the protocol (*request*, the customer has requested a quote, *accepted*, the customer has accepted the quote, etc.), but also special fluents to model *commitments* (and conditional commitments) among the roles [22,20]: $C(i, j, \alpha)$ says that role $i$ is committed to role $j$ to bring about $\alpha$. Furthermore, a *conditional commitments* $CC(i, j, \beta, \alpha)$ says that role $i$ is committed to role $j$ to bring about $\alpha$, if the condition $\beta$ is brought about.

The action theory $\Pi_{mr}$ consists of *action laws*, *causal laws*, *precondition laws*, and an *initial state*. *Action laws* $\mathcal{AL}_{mr}$ describe the effects of the execution of actions on the state. The action laws:

$$\Box_{mr}([\mathit{sendOffer}]_{mr}(\mathit{offer} \land CC(mr, ct, \mathit{accept}, \mathit{goods})))$$
$$\Box_{mr}(\mathit{request} \rightarrow [\mathit{sendOffer}]_{mr}\neg \mathit{request})$$

say that: when *mr* sends the quote for the good, then it commits to send the goods if *ct* accepts the request; and when *mr* sends the quote for the good, if there is a request, the request is cancelled.

*Causal laws* $\mathcal{CL}_{mr}$ are intended to expresses "causal" dependencies (or ramifications) among fluents. In this framework they are used to rule the dynamics of commitments. For instance, the *causal law*:

$$\Box_{mr}(\bigcirc_{mr}\alpha \rightarrow \bigcirc_{mr}\neg C(i, j, \alpha))$$

says that a commitment to bring about $\alpha$ is cancelled when $\alpha$ holds. Other causal laws are needed for dealing with conditional commitments.

*Precondition laws* $\mathcal{PL}$ have the form: $\Box(\alpha \rightarrow [a]\bot)$, meaning that the execution of an action $a$ is not possible if $\alpha$ holds. The precondition law

$$\Box_{mr}(\neg \mathit{request} \rightarrow [\mathit{sendOffer}]_{mr}\bot)$$

says that an offer cannot be sent if a request has not been issued.

The *initial state* $\mathcal{IS}$ of the protocol defines the initial value of all the fluents. Here, we assume that the initial state is complete.

Action laws and causal laws describe the changes to the state. All other fluents which are not changed by action execution are assumed to persist unaltered to the next state. To cope with the *frame problem* [18] we use a completion construction *Comp*, which is applied given a domain description, introduces frame axioms in the style of the successor state axioms proposed by Reiter [19]. The completion construction *Comp* is only applied to the action laws and to the causal laws [13]. Thus $\Pi_{mr}$ is defined as $Comp(\mathcal{AL} \wedge \mathcal{CL}) \wedge \mathcal{PL} \wedge \mathcal{IS}$.

The second component $\mathcal{C}_{mr}$ of the domain description $P_{mr}$ defines constraints as arbitrary temporal formulas of DLTL$_{mr}$. For instance, to model the fact that the merchant cannot send more than one offer, we introduce the constraint: $\neg\Diamond_{mr}\langle sendOffer\rangle_{mr}\Diamond_{mr}\langle sendOffer\rangle_{mr}\top$.

We are interested in those execution of the purchase protocol in which all commitments of all the roles have been fulfilled. In particular, for each commitment $C(i, j, \alpha)$ of which the merchant is a debtor or a creditor (i.e., $i = mr$ or $j = mr$), we add in $\mathcal{C}_{mr}$ the constraint: $\Box_{mr}(C(i, j, \alpha) \rightarrow \Diamond_{mr}(\alpha \vee \neg C(i, j, \alpha)))$, meaning that a commitment has to be fulfilled unless it is cancelled.

Given $D_{mr} = (\Pi_{mr}, \mathcal{C}_{mr})$ as defined above, we let $P_{mr} = \Pi_{mr} \wedge \mathcal{C}_{mr}$. Once the protocols $P_{ct}$, $P_{mr}$ and $P_{bk}$ have been defined, the specification $Pu$ of the Purchase protocol can be given as: $Pu = P_{ct} \wedge P_{mr} \wedge P_{bk}$. The *runs* of the protocol are then defined to be the linear models of $Pu$. They are all the runs that can be obtained by interleaving the actions of the runs of $P_{ct}, P_{mr}$ and $P_{bk}$, while synchronizing on common actions. By projecting the runs of the protocol $Pu$ to the alphabets of the participating roles, we get runs of each role $P_{ct}, P_{mr}$ and $P_{bk}$. As the properties we will consider in this paper regard only the sequence of communicative action exchanged between agents, in the following, we will consider protocol runs as infinite sequences of actions, and disregard worlds.

Note that protocol runs are always infinite, as logic DLTL is characterized by infinite models and we assume that all agents have infinite runs. Therefore, infinite protocols can be easily modelled in our framework. For instance, to model the infinite protocol in which the customer repeatedly issues a request, we can add to the specification above the constraint: $\Box_{ct}\Diamond_{ct}\langle sendRequest\rangle_{ct}\top$, requiring that the message request is issued infinitely many times. When we want to model terminating protocols, as the Purchase protocol above, we assume the domain description of each role of the protocol to be suitably extended with an action $noop_i$ which does nothing and which can be executed forever after termination of the protocol. Hence, in the following, we will assume that, for all $i$, the *i-th* projection of a run $\sigma$ of a protocol $P$ is an infinite run $\sigma|_i$ of $P_i$.

A protocol specification similar to the one above has been used in [12] to deal with problem of service composition. As a difference, here we use the product version of DLTL so to specify the role of each role $P_i$ independently, while in [12] a global specification of the protocol was given, including all roles.

A protocol defined as above, is not guaranteed to be well behaved, and we must impose some constraints on its structure to guarantee that the different roles participating in the protocol interoperate. To define the interoperability of a set of roles, we will put conditions on the computations of the interacting roles, which can be described by making use of Büchi automata.

### 2.1 Automata Based Verification

As usual for LTL, a DLTL formula can be mapped into a Büchi automaton so that the accepting runs of the automaton correspond to the models of the formula.

We recall that a *Büchi automaton* has the same structure as a traditional finite state automaton, with the difference that it accepts infinite words. More precisely a Büchi automaton over an alphabet $\Sigma$ is a tuple $\mathcal{B} = (Q, \rightarrow, Q_{in}, F)$ where:

- $Q$ is a finite nonempty set of states;
- $\rightarrow \subseteq Q \times \Sigma \times Q$ is a transition relation;
- $Q_{in} \subseteq Q$ is the set of initial states;
- $F \subseteq Q$ is a set of accepting states.

Let $\sigma \in \Sigma^\omega$ be an infinite word on $\Sigma$. Then a run of $\mathcal{B}$ over $\sigma$ is a mapping $\rho : prf(\sigma) \rightarrow Q$ such that $\rho(\varepsilon) \in Q_{in}$ and $\rho(\tau) \xrightarrow{a} \rho(\tau a)$ for each $\tau a \in prf(\sigma)$.

The run $\rho$ is *accepting* iff $inf(\rho) \cap F \neq \emptyset$, where $inf(\rho) \subseteq Q$ is given by $q \in inf(\rho)$ iff $\rho(\tau) = q$ for infinitely many $\tau \in prf(\sigma)$. Informally, a run is accepting if it goes infinitely many times trough a final state.

As described in [15], the satisfiability problem for DLTL can be solved in deterministic exponential time, as for LTL. Given a domain description $(\Pi, \mathcal{C})$, a corresponding Büchi automaton can be obtained such that all runs accepted by the automaton represent runs of the protocol, and vice versa. An algorithm for constructing on-the-fly a Büchi automaton from a DLTL formula has been proposed in [10], by generalizing the tableau-based algorithm for LTL. In general, this Büchi automaton is non-deterministic.

Let $P = P_1 \wedge \ldots \wedge P_k$ be a protocol such that each role $P_i$ is specified by a nondeterministic Büchi automaton $\mathcal{M}_i = (Q_i, \rightarrow_i, Q_i^{in}, F_i)$, whose accepting runs provide all the correct executions of the role. We assume that these automata have been "pruned" by eliminating all the states which do not occur on any accepting run. This can be achieved by starting from the accepting states, and by propagating backwards the information on the states for which a path to an accepting state exists.

The interactions of $P_1, \ldots, P_k$ can be described by the runs of a product automaton $\mathcal{M}$ over $\tilde{\Sigma}$. The states of $\mathcal{M}$ will be k-tuples $\langle s_1, \ldots, s_k \rangle$ of states of the automata $\mathcal{M}_1, \ldots, \mathcal{M}_k$. Let $Q^{\mathcal{M}} = Q^1 \times \ldots \times Q^k$ be the set of global states of $\mathcal{M}$. The transitions of $\mathcal{M}$ correspond to the execution of shared actions. The i-local transition relations induce a global transition relation $\rightarrow^{\mathcal{M}} \subseteq Q^{\mathcal{M}} \times \Sigma \times Q^{\mathcal{M}}$ as follows:

$q \xrightarrow{a}_{\mathcal{M}} q'$ iff $q[i] \xrightarrow{a}_i q'[i]$, for each $i \in Ag(a)$

$$q[i] = q'[i], \text{ for each } i \notin Ag(a)$$

where $q[i]$ denotes the $i$th component of $q = \langle q_1, \ldots, q_k \rangle$, and $Ag(a)$ is the set of agents sharing action $a$. Moreover, $Q^{in} \subseteq Q_1^{in} \times \ldots \times Q_k^{in}$ is the set of global initial states of $\mathcal{M}$.

A run of $\mathcal{M}$ over $\sigma \in \Sigma^\infty$ is a mapping $\rho : prf(\sigma) \rightarrow Q^{\mathcal{M}}$ such that $\rho(\varepsilon) \in Q^{in}$ and $\rho(\tau) \xrightarrow{a}_{\mathcal{M}} \rho(\tau a)$ for each $\tau a \in prf(\sigma)$. The run is *accepting* if, for all $i = 1, \ldots, k$, $\sigma|_i$ is infinite and $\rho(\tau)[i] \in F_i$ for infinitely many $\tau \in prf(\sigma)$. The runs of $\mathcal{M}$ describe the interleaving of the executions of $P_1, \ldots, P_k$, synchronizing on common actions.

Observe that, in an accepting run of $\mathcal{M}$, each $\mathcal{M}_i$ is executing a loop containing at least an accepting state in $F_i$. We will call such a loop in $\mathcal{M}_i$ an *accepting loop for* $P_i$. Vice-versa, a loop in $\mathcal{M}_i$ which does not contain any state in $F_i$ will be called a *non accepting loop for* $P_i$. In the following section we introduce a notion of interoperability of a set of roles $P_1, \ldots, P_k$ by putting conditions on (accepting and non accepting) runs of $\mathcal{M}$ and of the $\mathcal{M}_i$'s.

## 3 Interoperability

Let us consider an alternative specification $P'_{ct}$ of the customer role, according to which, after sending a request, the customer has to wait for an offer and it does not expect to receive from the merchant the answer "goods is not available". In such a case, the customer role $P'_{ct}$ would not interact properly with the merchant role $P_{mr}$ as defined in the previous section. If the merchant, after receiving a request from the customer, chooses to reply with *sendNotAvail*, the computation gets stuck, as the customer role $P'_{ct}$ cannot receive this message.

This example shows that, as the different roles of the protocol are defined separately, some requirement is needed to guarantee that such roles interact properly, so that the protocol as a whole is well defined. In particular, the interaction of the roles in the protocol should not produce deadlock. Similarly, we want to avoid infinite executions in which some role $P_i$ of the protocol is not executing an accepting run as, either, from some point onwards, $P_i$ does not execute any action, or $P_i$ is executing infinitely many actions, but on a non accepting run for $P_i$.

In the following, given the roles $P_1, \ldots, P_k$ of a protocol, as introduced in Section 2, we say that roles $P_1, \ldots, P_k$ are *interoperable* when they are free of choosing their actions at each step avoiding deadlock and non accepting executions. Let us point out that here we are considering in two different ways the nondeterministic choices concerning emissions (the customer can accept or refuse an offer) and those concerning receptions (the customer can receive the messages *sendOffer* or *sendNotAvail*). As usual in agent and web service applications, we assume that, in the first case, the choice is internal to the role (internal non determinism), while, in the second case, the choice is external to the agent and depends on the environment, namely on the interleaving of actions of partner agents (external non determinism). Hence, we postulate that a role can choose which message to send among the messages it can send in a state (the customer can decide whether to accept or refuse an offer), but that it cannot chose which message to receive among the messages he can receive in a state (the customer waits for *sendOffer* or *sendNotAvail*, but it cannot choose which one it will receive).

We can now define a notion of interoperability. We will denote by $m(i,j)$ the communicative action $m$ sent from $i$ to $j$. Let $\pi_i = q_0 \stackrel{a_1}{\to} q_1 \stackrel{a_2}{\to} \ldots \stackrel{a_v}{\to} q_v$ be the prefix of a run of $\mathcal{M}_i$. To model the fact that each $P_i$ must be able to choose which action to execute after $\pi_i$, we introduce a function $choice(P_i, \pi_i)$, whose value is defined as follows: either $choice(P_i, \pi_i) = m(i,j)$, where $m(i,j)$ is a send action that can be executed after $\pi_i$ on an accepting run of $\mathcal{M}_i$ (i.e., there is an accepting run of $P_i$ with prefix $q_0 \stackrel{a_1}{\to} q_1 \stackrel{a_2}{\to} \ldots \stackrel{a_v}{\to} q_v \stackrel{m(i,j)}{\to} q_{v+1}$); or the value $choice(P_i, \pi_i) = receiveR$, where $R = \{m_1(j_1,i), \ldots, m_n(j_n,i)\}$ contains all the receive actions that can be executed after $\pi_i$ on an accepting run of $\mathcal{M}_i$. In the last case, $P_i$ expects to receive a message from another agent after $\pi_i$ but it doesn't know which one it will receive among those messages $\{m_1(j_1,i), \ldots, m_n(j_n,i)\}$ it is able to receive after $\pi_i$. Observe that the choice of agent $P_i$ in $\pi_i$ may depend on the state $q_v$ but also on the sequence of actions $a_1, a_2, \ldots, a_v$ executed by $P_i$ up to $q_v$.

While we have assumed that agents can choose among the messages they can send, we have postulated that they cannot decide which message they will receive among those they are able to receive in a given state.

As a matter of notation, in the following, we say that $\sigma$ is an *execution of* $P$ when there is a (not necessarily accepting) run $q_0 \stackrel{a_1}{\to} q_1 \stackrel{a_2}{\to} \ldots$ of $\mathcal{M}$ over $\sigma = a_1, a_2, \ldots$ (representing a possible computation of $P_1, \ldots, P_k$). We say that $\pi$ is a *finite execution of* $P$ when $\pi$ is a finite prefix of an execution of $P$. We say that $\sigma$ is an *accepting run of* $P$ when there is an accepting run of $\mathcal{M}$ over $\sigma$.

We say that $\sigma$ is an execution of $P_1 \ldots P_k$ *respecting the function "choice"* if $\sigma$ is an execution of $P_1 \ldots P_k$ such that, for each prefix $\pi m(i,j)$ of $\sigma$, it holds that $choice(P_i, \pi|_i) = m(i,j)$ and $choice(P_j, \pi|_j) = receiveR$, with $m(i,j) \in R$.

We say that *a choice function is fair for* $P_i$ if there is no execution of $P_i$ respecting the choice function, in which $P_i$ executes infinitely many times a non-accepting loop of $\mathcal{M}_i$, although there is a send action $m(i,j)$ that $P_i$ can execute in some state of the loop leading outside the loop. A choice function is *fair* if it is fair for all the $P_i$'s. We say that $\sigma$ is a *fair execution of* $P_1 \ldots P_k$ if $\sigma$ is an execution of $P_1 \ldots P_k$ respecting a fair choice function.

In essence, according to a fair choice function, role $P_i$ cannot choose to execute infinitely many times a non accepting loop, if it can exit the loop by executing a send action: $P_i$ cannot be willing to execute infinitely many times a non-accepting loop. The interoperability of a set of roles is defined as follows:

**Definition 1.** $P_1 \ldots P_k$ *are* interoperable *if the following conditions hold:*

*(i) For any function* choice *and any finite execution $\pi$ of $P_1 \ldots P_k$, there exists an action $m(i,j)$, such that $choice(P_i, \pi|_i) = m(i,j)$ and $choice(P_j, \pi|_j) = receiveR$ (with $m(i,j) \in R$), and $m(i,j)$ is the first action executed after $\pi$ on an execution $\sigma$ of $P_1 \ldots P_k$ with prefix $\pi$*[1]*.*

*(ii) For each fair* choice*, each infinite execution $\sigma$ of $P_1 \ldots P_k$ respecting* choice*, is an accepting run of $P_1 \ldots P_k$.*

[1] Remember that $\pi|_i$ is the projection of $\pi$ on the alphabet of $P_i$.

According to the above definition, any prefix obtained by the execution of $P_1, \ldots, P_k$ can be extended by executing a new action according to the *choice* function. In particular, each agent can choose which action it wants to execute at each stage of the computation and, whatever the choice might be, the computation does not get stuck (condition(i)) and, eventually, each agent can execute his choice in an accepting run (condition (ii)).

Observe that the choice of a role $P_i$ after a sequence $\pi$ of execution steps only depends on the actions of $P_i$ up to that point (namely, on $\pi|_i$). Hence, the choice of $P_i$ is left unchanged by the execution of communicative actions not involving $P_i$. As a consequence, the choice of agent $P_i$ remains unchanged until eventually it is fulfilled by the execution of the chosen action.

The conditions above guarantee that the choice of an agent cannot be delayed forever by the choices of other agents. The following proposition follows easily from the definition of interoperability above:

**Proposition 1.** *(a) For each fair choice functions, for each finite executions $\pi$ of $P_1 \ldots P_k$, if choice$(P_i, \pi|_i) = m(i,j)$ (or choice$(P_i, \pi|_i) =$ receiveR), then there is an accepting run $\sigma$ of $P_1 \ldots P_k$ respecting the choice function, such that $\sigma$ has prefix $\pi\pi' m(i,j)$ (respectively, $\pi\pi' m(j,i)$, with $m(j,i) \in R$), where $\pi'$ does not contain actions of $P_i$.*
*(b) If $P_1 \ldots P_k$ are interoperable, then $P_1 \ldots P_k$ have an accepting run.*

*Proof.* By condition (i), the finite execution $\pi$ can be extended to an infinite execution $\sigma$ respecting the fair choice function. By condition (ii) the fair execution $\sigma$ respecting the fair choice function is an accepting run of $P_1 \ldots P_k$. As $P_i$ must execute infinitely many actions on $\sigma$, it will eventually execute an action after $\pi$ according to his choice in $\pi$. Namely, $P_i$ will eventually execute $m(i,j)$ (respectively, $m(j,i)$) as its first action after $\pi$. The proof of point (2) is similar.

Let us consider the example in Figure 1. Role $P_1$ can repeatedly either send message $m(1,2)$ to $P_2$ or message $m(1,3)$ to $P_3$. Role $P_1$ can go on choosing to send $m(1,2)$ to $P_2$, so that $P_3$ is not executing any action. This choice is fair. $P_2$ and $P_3$ have the only (fair) choice of executing the receive $m(1,2)$ and the receive $m(1,2)$, respectively. $P_1, P_2, P_3$ are not interoperable. According to the above fair choices, $P_1$ and $P_2$ can go on exchanging message $m(1,2)$, and they

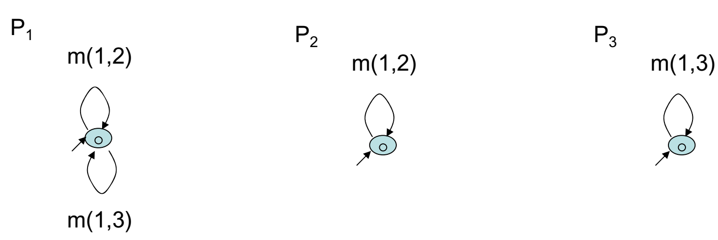


**Fig. 1.**

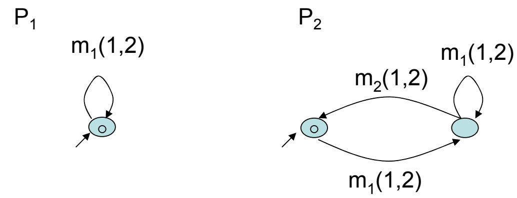


**Fig. 2.**

produce an execution $\sigma$ which is an accepting run of $P_1$ and of $P_2$, but is not an accepting run of $P_3$ ($P_3$ does not execute any action on $\sigma$). $\sigma$ is a fair execution of $P_1, P_2, P_3$, but it is not a run of $P_1, P_2, P_3$. Condition (ii) fails.

As a different example, let us consider the example in Figure 2. The runs of $P_2$ are those sequences obtained by repeating infinitely many times: a (nonempty) finite sequence of actions $m_1(1,2)$ followed by an action $m_2(1,2)$. $P_1$ and $P_2$ can go on by exchanging the message $m_1(1,2)$, thus producing a fair execution $\sigma'$ (in each state, $P_2$ has the fair choice of executing a receive, but it cannot control which message is received). While for $P_1$ $\sigma'$ an accepting run, for $P_2$ it is not. Although $P_2$ executes infinitely many actions in $\sigma'$, it does not execute a run. $P_1$ and $P_2$ are not interoperable, as condition (ii) is not satisfied.

## 4 Conformance

Let $A_1, \ldots, A_k$ be a set of agents. The specification of each agent $A_i$ can be either given through a logical specification as the one introduced in Section 2 for roles, or by introducing the automaton describing the possible behaviors of the agent. We assume that the actions of each agent $A_i$ are deterministic and that the automaton describing its possible behaviors is a deterministic Büchi automaton. The notions of *execution of* $A_1, \ldots, A_k$ and of *accepting run of* $A_1, \ldots, A_k$ are defined as for $P_1, \ldots, P_k$ in Section 3.

Given a protocol $P = P_1 \wedge \ldots \wedge P_k$ with $k$ roles, we define the *conformance of a set of agents* $A_1, \ldots, A_k$ *with* $P$, as follows:

**Definition 2.** *Agents* $A_1, \ldots, A_k$ *are* conformant *with* $P$ *if:*
*(a)* $A_1, \ldots, A_k$ *are interoperable, and*
*(b) all accepting runs of* $A_1, \ldots, A_k$ *are accepting runs of the protocol* $P$*.*

In this section we want to introduce a notion of *conformance of a single agent* $A_i$ *with the protocol* $P$, so that the conformance of each $A_i$, proved independently, guarantees the conformance of the overall set of agents $A_1, \ldots, A_k$ with $P$, according to Definition 2.

Given the definition of interoperability given in the previous section, this notion of conformance can be based on the policy: *less emissions and more receptions* [4,2,11]. Consider, for instance, a customer agent $A_{ct}$ whose behavior differs from that of the role "customer" of protocol $Pu$ as follows: whenever it receives an offer from the merchant, it always accepts it; after accepting the offer it expects to receive from $P_{mr}$ either *sendGoods* or *cancelDelivery*. Although the behavior of $A_{ct}$ and that of the corresponding role of the protocol are different, we could consider however the agent to be conformant with the protocol, since the customer can choose which messages to send, and thus it is not forced to send all the messages required by the protocol. Also, the agent can receive more messages than those required by the protocol, since these receptions will never be executed.

Let $\mathcal{M}_i$ be the automaton specifying role $P_i$ and $\mathcal{M}_i^A$ the automaton specifying agent $A_i$. We can formulate the policy "less emissions and more receptions" with the following condition. Let $\pi$ be a finite execution common to $A_i$ and $P_i$:

(C1) [Correctness of emissions of $A_i$]
If $\pi m(i,j)$ is a finite execution of $A_i$ (where $m(i,j)$ is a send action of $A_i$), then $\pi m(i,j)$ is a finite execution of $P_i$.

(C2) [Completeness of receptions of $A_i$]
If $\pi m(j,i)$ is a finite execution of $P_i$ (where $m(j,i)$ is a receive action of $P_i$), then $\pi m(j,i)$ is a finite execution of $A_i$.

Unfortunately, the policy "less emissions and more receptions" only works for two-party protocols, as shown in the next example.

*Example 2.* Consider protocol $Pu$. The customer *ct*, at some point, may accept the offer of $mr$ and require *bk* to send the payment to $mr$. Assume that $mr$ has the requirement that it can receive the payment from *bk* only after it has received the acceptance of the offer from *ct*. Namely, the specification of $P_{mr}$ contains the constraint:

(*) $\Box_{mr}(\neg accepted \rightarrow [sendPayment]_{mr}\bot)$.

According to the protocols of *ct* and *bk* the message $sendPayment$ can be sent from *bk* to $mr$ either before or after the message *sendAccept* is sent from *ct* to $mr$. It is clear that, although *ct* and *bk* do not put constraints on the order in which they send the acceptance of the offer and the payment to $mr$, in the overall protocol $Pu$ they are forced to respect the constraint of the merchant, and only the runs in which *sendAccept* is executed before $sendPayment$ are accepted as runs of $Pu$.

Let us now consider an agent $A_{mr}$, playing the role of the merchant, whose behavior is the following: either it receives a message *sendAccept* followed by a message $sendPayment$, or receives a message $sendPayment$ followed by a message *sendAccept*. Agent $A_{mr}$ allows for more receptions with respect to its role $P_{mr}$ (in fact, $A_{mr}$ receives the additional message $sendPayment$ followed by a message *sendAccept*). When $A_{mr}$ interacts with a customer and a bank agents behaving as stipulated by $P_{ct}$ and $P_{bk}$, it may produce an execution in which $sendPayment$ comes before *sendAccept*, which is not a run of protocol $Pu$ (as it does not satisfy the constraint (*) above). Conversely, if we consider a variant $Pu'$ of the $Pu$ protocol in which the roles of the bank $P'_{bk}$ and of the customer $P'_{ct}$ coordinate their executions so that they execute *sendAccept* before *sendPayment*, the interaction of agent $A_{mr}$ with the other roles cannot produce the unwanted execution in which *sendPayment* comes before *sendAccept*.

In this example, $A_{mr}$ can be regarded to be non-conformant with protocol $Pu$, but conformant with the protocol $Pu'$, although $P_{mr}$ is the same in both protocols. The case in which $A_{mr}$ is non-conformant with $Pu$, is similar to the example discussed in [17], where the problem of conformance checking is analyzed for models of asynchronous message passing software. The solution adopted in [17] is that of requiring that an agent $A_i$ cannot do more receptions than those established by protocol $P_i$, so that: $A_i$ can do less emission and exactly the same receptions as stated by $P_i$. In the example above, this would correspond to take $A_{mr}$ as being non-conformant. We believe that this policy is too restrictive: as we see from Example 2, the conformance of $A_{mr}$ depends on the overall protocol, including other roles. In the following, we propose a definition of the conformance of an agent $A_i$ with respect to the overall protocol $P$, rather than to its role $P_i$. We are ready to admit additional receptions in $A_i$, if we are sure that such receptions cannot give rise to unwanted executions when $A_i$ interacts with agents respecting protocol $P$.

In the following, besides referring to the executions and runs of a protocol $P$, we need to refer to the executions and runs of an agent $A_i$ in the context of the protocol $P$. We will denote by $P[A_i]$ the set of roles/agents $P_1, \ldots, P_{i-1}, A_i, P_{i+1}, \ldots, P_k$. $P[A_i]$ represents the protocol obtained from $P$ by replacing role $P_i$ with agent $A_i$. Also, we will refer to the executions (accepting runs) of $P_1, \ldots, P_{i-1}, A_i, P_{i+1}, \ldots, P_k$ as executions (accepting runs) of $P[A_i]$.

To guarantee that an agent $A_i$ is conformant with a protocol $P$, we need to introduce, besides (C1) and (C2) above, further conditions which ensure that $A_i$ interacts properly with the other roles in the protocol $P$:

(C3) *[Interoperability]* $P[A_i]$ interoperate
(C4) *[Correctness of the receipts of $A_i$ in the context of $P$]* All accepting runs of $P[A_i]$ are accepting runs of $P$.

Condition (C3) says that $P[A_i]$ is interoperable, that is $A_i$ interacts with with $P_1, \ldots, P_{i-1}, P_{i+1}, \ldots, P_n$ so that each role can make its choices without the computation getting stuck or ending up in non accepting loops.

Condition (C4) requires that the executions of $A_i$ are *correct* when $A_i$ is interacting with other agents respecting the protocol $P$. In particular, this ensures that although $A_i$ can execute more receptions than $P_i$, such additional receptions are not executed when $A_i$ interacts with the other roles in $P$.

Observe that condition (C4) can be equivalently expressed in the logic by saying that the formula $P[A_i] \rightarrow P$ has to be valid.

According to the above definition of conformance, the merchant agent $A_{mr}$ in Example 2 is conformant with $Pu'$, while $A_{mr}$ is non-conformant with protocol $Pu$. In fact, although $Pu[A_{mr}]$ is interoperable (and, in particular, $A_{mr}$ can interact with other agents executing the protocol without getting stuck), there is a run of $Pu[A_{mr}]$, in which *sendPayment* comes before *sendAccept*, that is not a run of $Pu$ (and therefore (C4) is violated). This last case shows that the interoperability of $A_i$ with the other roles in $P$ is not sufficient to guarantee the correctness of the resulting runs with respect to $P$.

Are conditions (C1) to (C4) sufficient to guarantee the conformance of an agent with a protocol? What we expect is that, given an interoperable protocol $P$ and a set of agents $A_1, \ldots, A_k$, if each agent $A_i$ is conformant with $P$ (according to (C1)...(C4)) then the agents $A_1, \ldots, A_k$ interoperate and their accepting runs are runs of $P$.

Consider the roles $P_1$ and $P_2$ and the agents $A_1$ and $A_2$ in Figure 3. $P_1$ and $P_2$ interoperate. $A_1$ has the same receptions and less emissions than $P_1$ (the send action $m_3(1,2)$ is not present in $A_1$). $A_1$ is conformant with $P$ (according to conditions (C1),..,(C4)). In particular, $A_1$ and $P_2$ interoperate: for any fair choice function, $P_2$ cannot produce the infinite execution $\sigma = m_1(1,2), m_2(2,1), m_1(1,2), m_2(2,1), \ldots$, as $P_2$ must eventually execute the send action $m_4(2,1)$. Similarly, $A_2$ is conformant with $P$ and it has the same receptions and less emissions than $P_1$ (it misses the send action $m_4(2,1)$). However, $A_1$ and $A_2$ do not interoperate: $A_1$ and $A_2$ have the only choice of executing $\sigma$ and this is a fair choice for $A_1$ and $A_2$. However, $\sigma$ is not a run of $A_1, A_2$, and this violates interoperability condition (ii).

Observe that the choice of executing $\sigma$ is fair for $A_1$, as $A_1$ has not the choice of executing any action to exit the non accepting loop in $\sigma$. Instead, the execution $\sigma$ is not fair for $P_1$, as $P_1$ can choose to exit the non accepting loop by executing the send action $m_3(1,2)$. We introduce the following condition:

(C5) For each $\sigma$ execution of both $A_i$ and $P_i$, if $\sigma$ is a fair execution of $A_i$, then $\sigma$ is a fair execution of $P_i$.

Condition (C5) is violated by $A_1$ (and by $A_2$) in Figure 3. The choice of executing $\sigma$ is fair for $A_1$, as there is no send action that agent $A_1$ can execute to exit from the non accepting loop. Instead, the choice of executing $\sigma$ is not fair for $P_1$ which can execute action $m_3(1,2)$ to exit from the non accepting loop. The verification of condition ($C5$) requires reasoning on non accepting loops in the automata of $A_i$ and $P_i$. The idea is that, although $A_i$ can contain less

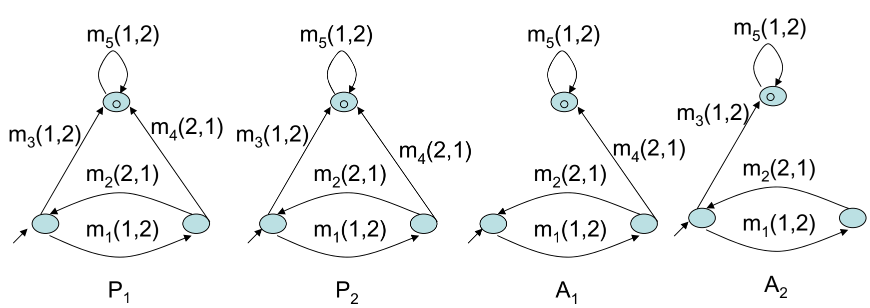


**Fig. 3.**

emissions than $P_i$, it must contain al least those emissions allowing to go out from non accepting loop and to go on with an accepting run, if this is possible for $P_i$.

Finally, we introduce the following conditions to deal with infinite computations, which requires that, when $A_i$ and $P_i$ produce a common action sequence, then either they both accept it or they both do not accept it.

(C6) For all $\sigma$ that are both executions of $A_i$ and of $P_i$, $\sigma$ is an accepting run of $A_i$ if and only if $\sigma$ is an accepting run of $P_i$.

The notion of conformance of an agent with a protocol is then defined as follows:

**Definition 3.** *[Conformance of agent $A_i$ with protocol $P$] $A_i$ is* conformant *with a protocol $P = P_1 \wedge \ldots \wedge P_k$ if the conditions (C1),..., (C6) above are satisfied.*

Let $P = P_1 \wedge \ldots \wedge P_k$ be an interoperable protocol. $P$ has at least one accepting run. Let an execution of $A_1, \ldots, A_k$ be a run of the product automaton $\mathcal{A}$ defined as the product automaton $\mathcal{M}$ in Section 2.1. Let an accepting run of $A_1, \ldots, A_k$ be an accepting run of the product automaton $\mathcal{A}$. We can prove that, given $k$ agents $A_1, \ldots, A_k$, if each $A_i$ is conformant with $P$ according to Definition 3, then the accepting runs of $A_1, \ldots, A_n$ are accepting runs of protocol $P$ and $A_1, \ldots, A_n$ interoperate.

**Theorem 1.** *Let $A_1, \ldots, A_k$ be $k$ agents. If, for each $i = 1, \ldots, k$, agent $A_i$ is conformant with protocol $P$ according to Definition 3, then agents $A_1, \ldots, A_k$ are conformant with $P$ according to Definition 2.*

In order to prove the result above, we prove the following lemmas. Lemma 1 says that any finite execution $\pi$ of a set of agents $A_1, \ldots, A_k$, with each $A_i$ conformant with $P$, is a prefix of a run of $P$.

**Lemma 1.** *Assume that, for all $i = 1, \ldots, k$, agent $A_i$ is conformant with protocol $P = P_1 \wedge \ldots \wedge P_k$ according to Definition 3. If there is a finite execution $\pi$ of $A_1, \ldots, A_k$, then there is an accepting run $\rho$ of $P$ with prefix $\pi$.*

*Proof.* We prove that there is an accepting run of $P$ with prefix $\pi$ by induction on the length $l$ of $\pi$. If $\pi = \epsilon$, the theorem holds, as there exists an accepting run of $P$, by Proposition 1(b).

For the inductive case, let $\pi m(i,j)$ be an execution of $A_1, \ldots, A_k$ of length $l+1$. We show that there is an accepting run of $P$ with prefix $\pi m(i,j)$. By inductive hypothesis, there is an accepting run of $P$ with prefix $\pi$. As $A_i$ can execute action $m(i,j)$ after $\pi|_i$, by the correctness of the send of $A_i$ with respect to $P_i$, condition (C1), also $P_i$ can execute action $m(i,j)$ just after $\pi|_i$.

From the hypothesis we know that $A_j$ can receive $m(i,j)$ as its first action after $\pi$. As $P_i$ can emit $m(i,j)$ as its first action after $\pi$, $A_j$ and $P_i$ can synchronize to execute $m(i,j)$ after $\pi$ in $P[A_j]$. Hence, $\pi m(i,j)$ is a finite execution of $P[A_j]$. Let us consider any fair choice function such that the finite execution $\pi m(i,j)$

respects the choice function. By (C3) $P[A_j]$ interoperate and, by Proposition 1, $\pi m(i,j)$ can be extended to an accepting run $\rho$ of $P[A_j]$ respecting the choice function. By (C4), $\rho$ is also a run of $P$. Hence, there is an accepting run of $P$ with prefix $\pi m(i,j)$. □

Lemma 1 can be generalized to prove that, if there is a finite execution $\pi$ of $P[A_1,\ldots,A_h]$ (with $h \leq k$), then there is an accepting run $\rho$ of $P$ with prefix $\pi$. We can now prove the following interoperability result.

**Lemma 2.** *Assume that, for all $i = 1,\ldots,k$, agent $A_i$ is conformant with protocol $P$ according to definition 3. If the roles of $P$, $P_1 \ldots P_k$, are interoperable, then also $A_1 \ldots A_k$ are interoperable.*

*Proof.* Assume that $P_1 \ldots P_k$, are interoperable. To show that $A_1 \ldots A_k$ are *interoperable*, we have to prove that $A_1 \ldots A_k$ satisfy conditions (i) and (ii) of Definition 1.

Let us prove (i). Let $choice_A$ be any choice function for $A_1 \ldots A_k$, and let $\pi$ be any finite execution of $A_1 \ldots A_k$. We want to prove that there exists an action $m(i,j)$, such that $choice_A(A_i, \pi|_i) = m(i,j)$ and $choice_A(A_j, \pi|_j) = receiveR$ (with $m(i,j) \in R$), and $m(i,j)$ can be executed by $A_1 \ldots A_k$ just after $\pi$.

By Lemma 1, $\pi$ is prefix of an accepting run $\sigma$ of $P_1 \ldots P_k$. Therefore, for all $i$, $P_i$ has a run $\sigma|_i$ with prefix $\pi|_i$. We want to exploit the fact that $P_1 \ldots P_k$ are interoperable and show that $choice_A$ can be taken as the choice function for the $P_i$'s in $\pi$. Let $I$ be the set of all $i$ such that $choice_A(A_i, \pi|_i) = m(i,j)$, that is, all $i$ such that agent $A_i$ wants to execute a send after $\pi$.

Let $i \in I$. As $choice_A(A_i, \pi|_i) = m(i,j)$, then $A_i$ can execute $m(i,j)$ after $\pi|_i$ on some of its runs. By the correctness of the send of $A_i$ with respect to $P_i$, condition (C1), also $P_i$ can choose to execute the send action $m(i,j)$ after $\pi|_i$. Hence, we can take $m(i,j)$ as the choice of $P_i$ after $\pi$.

Let $j \notin I$, that is $choice_A(A_j, \pi|_j) = receiveR$. We want to show that also $P_j$ can choose to execute a receive action after $\pi$. As $\pi$ is a finite execution of $A_1 \ldots A_k$ and the prefix of an accepting run of $P_1 \ldots P_k$, then $\pi$ is a finite execution of $P[A_j]$. By (C3), $P[A_j]$ interoperate. Let us consider any fair choice function $ch'$ for $P_1 \ldots P_k$ such that $ch'(A_j, \pi|_j) = choice_A(A_j, \pi|_j) = receiveR$. By interoperability of $P[A_j]$ there is an accepting run $\rho$ of $P[A_j]$ with prefix $\pi\pi' m(i,j)$, where $m(i,j) \in R$ and $\pi'$ does not contain actions of agent $j$. By (C4) $\rho$ is an accepting run of $P$. As $P_i$ executes the receive $m(i,j)$ as its first action after $\pi$ in $\rho$, $P_i$ can choose to execute a receive after $\pi$.

Now, let $choice_P$ be a choice function for $P_1 \ldots P_k$, such that $choice_P(P_i, \pi|_i) = choice_A(A_i, \pi|_i)$. By the interoperability of $P_1 \ldots P_k$ there is an action $m(i,j)$ which can be executed by the $P_i$'s after $\pi$ according to the choice function $choice_P$. Action $m(i,j)$ is the choice of agent $A_i$ after $\pi$, so that $A_i$ can send $m(i,j)$ after $\pi$. Since $P_j$ receives message $m(i,j)$ after $\pi$, by the conformance of $A_j$ to $P$, condition (C2), also agent $A_j$ can receive message $m(i,j)$ after $\pi$. Hence, $m(i,j)$ can be executed by $A_1 \ldots A_k$ after $\pi$ according to the choice function $choice_A$. This concludes the proof of point (i).

Let us prove (ii). We have to prove that, for each fair *choice*, each infinite execution $\sigma$ of $A_1, \ldots, A_k$ respecting *choice*, is an accepting run of $A_1, \ldots, A_k$.

Let $\sigma$ be an infinite execution of $A_1, \ldots, A_k$ respecting *choice*. It is easy to see that $\sigma$ is also an execution of $P$. In fact, by Lemma 1 we know that each prefix $\pi$ of an execution of $A_1, \ldots, A_k$ is also a prefix of a run of $P$. Hence, $\sigma$ is also an execution of $P_1, \ldots, P_k$. Given condition (C5), the *choice* function, which is fair for $A_1, \ldots, A_k$, must also be a fair choice function for $P_1, \ldots, P_k$. As $P$ is interoperable and the *choice* function is fair for $P_1, \ldots, P_k$, by the interoperability condition (ii), $\sigma$ is an accepting run of $P$. Observe that, for all $i$, $\sigma|_i$ is both an execution of $A_i$ and of $P_i$. By condition (C6), as $\sigma|_i$ is an accepting run of $P_i$, $\sigma|_i$ must also be an accepting run of $A_i$. It follows that $\sigma$ is an accepting run of $A_1, \ldots, A_k$. □

**Lemma 3.** *Assume that, for all $i = 1 \ldots k$, agent $A_i$ is conformant with protocol $P$ according to definition 3. If $\sigma$ is an accepting run of $A_1 \ldots A_k$, then $\sigma$ is an accepting run of $P$.*

*Proof.* By induction on $h$, we prove that: If $\sigma$ is an accepting run of $P[A_1, \ldots, A_h, A_{h+1}]$, then $\sigma$ is an accepting run of $P[A_1, \ldots, A_h]$. For $h = 0$, by (C4), if $\sigma$ is an accepting run of $P[A_1]$, then $\sigma$ is an accepting run of $P$.

For $h > 0$, assume that $\sigma$ is an accepting run of $P[A_1, \ldots, A_h, A_{h+1}]$ and that $\sigma$ is not an accepting run of $P[A_1, \ldots, A_h]$. It must be that $\sigma|_i$ is an accepting run of $A_{h+1}$, while $\sigma|_i$ is not an accepting run of $P_{h+1}$.

Observe that, by (C1), all the send actions of $A_{h+1}$ in $\sigma$ are correct and can be executed by $P_{h+1}$, but $A_{h+1}$ could receive a message $m(j, h+1)$ in $\sigma$ which cannot be received by $P_{h+1}$. If this is not the case, $\sigma$ is an execution of $P[A_1, \ldots, A_h]$ and, by (C6), $\sigma$ is an accepting run of $P[A_1, \ldots, A_h]$.

If there is a receive $m(j, h+1)$ in $\sigma$ which is not executable by $P_{h+1}$, then there must be a prefix $\pi m(j, h+1)$ of an execution of $P[A_1, \ldots, A_h, A_{h+1}]$ which is not a prefix of an accepting run of $P$ (as it is not a prefix of an accepting run of $P_i$). This contradicts Lemma 1. □

Observe that, although the interoperability of $A_i$ with $P$ (condition (C3)), guarantees that $A_i$ is able to receive the messages sent to it from the other roles in $P$, it does not enforce condition (C2). If (C2) were omitted in Definition 3, Theorem 1 would not be provable. In fact, in such a case, it might occur that, although agent $A_1$ is interoperable with other roles in $P$ and the same holds for agent $A_2$, the two agents $A_1$ and $A_2$ do not interoperate with each other and with the other roles in $P$.

Let, for instance, $P$ be an interoperable protocol with 4 roles, where each role has the following runs:

$P_1 : \{m(1,3), m(1,4)\}$
$P_2 : \{m(2,4), m(2,3)\}$
$P_3 : \{m(1,3), m(2,3) + m(2,3), m(1,3)\}$
$P_4 : \{m(1,4), m(2,4) + m(2,4), m(1,4)\}$

Let us consider now the agents $A_1, A_2, A_3, A_4$, such that $A_1 = P_1$, $A_2 = P_2$, while $A_3$ and $A_4$ have the following runs:

$$A_3 : \{m(2,3), m(1,3)\} \qquad\qquad A_4 : \{m(1,4), m(2,4)\}$$

It can be seen that $A_3$ satisfies conditions (C1) and (C3), but not (C2), and the same holds for $A_4$. The agents $A_1, A_2, A_3, A_4$ do not interoperate, since $A_1$ and $A_2$ must emit respectively $m(1,3)$ and $m(2,4)$ as their first message, but none of them can be received by $A_3$ or $A_4$.

Observe that some conditions in the definitions of interoperability and conformance can be easily verified, as for instance, the interoperability condition (i), which requires to check some conditions on all the states of the product automaton: from each state it must always be possible to execute some action according to the choices of the agents. This verification requires polynomial time in the size of the product automaton $\mathcal{M}$. However, the verification of other conditions, like the interoperability condition (ii) and of conditions (C4) and (C5) is rather complex. For instance, the verification of condition (ii) requires to execute a check on all loops in the automaton $\mathcal{M}$, that are non accepting loops for some role $P_i$. Also, the verification of condition (C4), that all the runs of $P[A_i]$ are runs of $P$, requires to check language inclusion between two non deterministic $\omega$ automata. This problem is, in the general case, PSPACE-hard [7].

We may wonder whether the addition of simplifying assumption may lead to a simplified definition of the notions of conformance and interoperability. Several simplifications naturally arise: the assumption that the agents have the same receptions as the corresponding roles; the assumption that the role and agent automata are deterministic; the assumption that all protocol are finite and that finite automata can be used rather then Büchi automata to model protocols.

Concerning the first simplification, if agent $A_i$ has that same receptions as $P_i$, conditions (C1) and (C3) can be replaced by the single condition (C1') All the runs of $A_i$ are runs of $P_i$ stating that both emissions and receptions of $A_i$ are correct with respect to $P_i$. Verifying this condition, requires to verify the validity of the entailment $A_i \rightarrow P_i$, which for DLTL (as for LTL) is a problem in PSPACE-hard. With this simplification however condition (C5) is still needed to guarantee interoperability, as shown, for instance by the example in Figure 3.

If we restrict our consideration to protocols described by deterministic Büchi automata, rather than to non deterministic ones, the verification of some condition becomes simpler (as, for instance, the verification of (C4)), although the conditions for interoperability and conformance remain unaltered.

In the case we are considering finite state automaton, some conditions, like (C6), are not needed. Also, the interoperability conditions (i) and (ii) gets simplified. However, to guarantee interoperability of the agents which are individually conformant with the protocol, some condition playing the role of (C5) is still needed, as shown by the example in Figure 3, which can be easily adjusted for finite state automata.

## 5 Related Work

The paper deals with the problem of agents conformance with multiparty protocols. The notion conformance we have introduced guarantees the interoperability of a set of agents which are conformant with the protocol. We have assumed that the specification of the protocol is given in a temporal action logic and we have shown that the verification of conformance can be done independently for each agents, by making use of automata-based techniques. The proposed approach deals with both terminating and non-terminating protocols.

The paper generalizes the proposal in [11], where a notion of conformance for two-party protocols has been defined based on the policy "less emissions and more receptions". Such policy, however, is not sufficient to guarantee stuck-freeness in the multiparty case. Indeed, the notion of conformance we propose here requires in addition: 1) the correctness of the receipts an agent can do, when it is interacting with other roles of the protocol, and 2) the interoperability of the agent with the other roles of the protocol. The notion of conformance proposed here is also stronger that the notion of compliance in [13]. There, an agent $A_i$ is said to be compliant with a protocol $P$ if, in all interactions of $A_i$ with other roles in the protocol, $A_i$ satisfies its commitments and permissions. This condition essentially corresponds to the correctness condition (C2) in Definition 3. The notion of compliance in [13] does not guarantee stuck-freeness.

Several other proposals have been put forward in the literature for dealing with agent and agent conformance and interoperability.

In [4], several notions of *compatibility* and *substitutability* among agents have been analyzed, in which agents are modelled by Labelled Transition Systems, communication is synchronous, and models are deterministic. Substitutability is related to the notion of conformance. [4] introduces two distinct notions of substitutability (related with conformance): a first one, based on the policy "less emissions and more receptions", which does not preserves deadlock-freeness, and a second more restrictive one requiring "the same emissions and receptions". Agent executions are always terminating.

In [2] an automata based approach is used for conformance verification, by taking into account the asymmetry between messages that are sent and messages that are received. Agents and protocols are represented as deterministic finite automata, and protocols have only two roles. The approach has been extended to the multiparty case in [3], which also accounts for the case of nondeterministic agents and roles producing the same interactions but having different branching structures. Such a case cannot be handled in the framework in [2] as well as in our framework, due to the fact that our approach is based on a trace semantics. A similar approach is also used in [1], where an abductive framework is used to verify the conformance of agents to a choreography with any number of roles. As a difference with the above proposals, our proposal deals with protocols with infinite runs and guarantees stuck-freeness also in the multiparty case.

In [17] a notion of conformance is defined to check if an implementation model $I$ conforms with a signature $S$, in the case both $I$ and $S$ are CCS processes, and communication is asynchronous. The policy "less emissions and the same receptions" is introduced to guarantee stuck-freeness. Our approach provides a solution to guarantees stuck-freeness without requiring an agent to have the same receptions as its role in the protocol.

The notions of conformance, coverage and interoperability are defined in [5]. A distinctive feature of that formalization is that the three notions are orthogonal to each other. Conformance and coverage are based on the semantics of runs and relies on the notion of run subsumption, concerning the single agent and its role in the protocol. Interoperability among agents is based upon the idea of blocking and depends on the computation that the agents can jointly generate. The paper only considers two-party protocols and agents with finite runs.

In [6] a notion of constitutive interoperability is proposed, which "abstracts from the process-algebraic notion of interoperability and makes commitment alignment the sole criteron" so to capture the business meaning of the business processes. As said in [6], this notion is complementary to a notion of regulative interoperability, which takes into consideration message order, occurrence and data flow. The work we have presented here falls within the context of regulative interoperability.

In [16] Web Services are modelled as MAS and model checking is used for verifying temporal epistemic properties of OWL specifications. An OWL specification of a service is mapped to ISPL (the language of the model checker MCMAS), and a coloring of states as compliant (green) or non compliant (red) is assumed for the verification. The paper does not provide formalization of contracts (or protocols) and it does not provide techniques for automatically computing, for given a service, the states which are compliant or non compliant with a contract.

[8] focuses on the realizability problem of a framework for modeling and specifying the behaviors of *reactive electronic services*. In that framework, services communicate by asynchronous message passing, and are modeled by means of Büchi automata. The authors show that not every conversation protocol is realizable in the framework, and give some realizability conditions and show that each conversation protocol satisfying those conditions is realizable.

## Acknowledgement

The work was partially supported by Regione Piemonte, Project ICT4LAW.

## References

1. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Montali, M.: An abductive framework for a-priori verification of web agents. In: Principles and Practice of Declarative Programming (PPDP 2006). ACM Press, New York (2006)
2. Baldoni, M., Baroglio, C., Martelli, A., Patti, V.: Verification of protocol conformance and agent interoperability. In: Toni, F., Torroni, P. (eds.) CLIMA 2005. LNCS, vol. 3900, pp. 265–283. Springer, Heidelberg (2006)
3. Baldoni, M., Baroglio, C., Martelli, A., Patti, V.: A Priori Conformance Verification for Guaranteeing Interoperability in Open Environments. In: Dan, A., Lamersdorf, W. (eds.) ICSOC 2006. LNCS, vol. 4294, pp. 339–351. Springer, Heidelberg (2006)

4. Bordeaux, L., Salaün, G., Berardi, D., Mecella, M.: When are two web-agents compatible, VLDB-TES (2004)
5. Chopra, A.K., Singh, M.P.: Producing Compliant Interactions: Conformance, Coverage, and Interoperability. In: Baldoni, M., Endriss, U. (eds.) DALT 2006. LNCS, vol. 4327, pp. 1–15. Springer, Heidelberg (2006)
6. Chopra, A.K., Singh, M.P.: Constitutive Interoperability. In: Proc. of the 7th Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), pp. 797–804 (2008)
7. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. MIT Press, Cambridge (2000)
8. Fu, X., Bultan, T., Su, J.: Conversation protocols: a formalism for specification and verification of reactive electronic services. Theor. Comput. Sci. 328(1-2), 19–37 (2004)
9. Giordano, L., Martelli, A., Schwind, C.: Verifying Communicating Agents by Model Checking in a Temporal Action Logic. In: Alferes, J.J., Leite, J. (eds.) JELIA 2004. LNCS (LNAI), vol. 3229, pp. 57–69. Springer, Heidelberg (2004)
10. Giordano, L., Martelli, A.: Tableau-based Automata Construction for Dynamic Linear Time Temporal Logic. Annals of Mathematics and Artificial Intelligence 46(3), 289–315 (2006)
11. Giordano, L., Martelli, A.: Verifying Agent Conformance with Protocols Specified in a Temporal Action Logic. In: Basili, R., Pazienza, M.T. (eds.) AI*IA 2007. LNCS (LNAI), vol. 4733, pp. 145–156. Springer, Heidelberg (2007)
12. Giordano, L., Martelli, A.: Web Service Composition in a Temporal Action Logic. In: 4th Int. Workshop on AI for Service Composition, AICS 2006, Riva del Garda, August 28 (2006)
13. Giordano, L., Martelli, A., Schwind, C.: Specifying and Verifying Interaction Protocols in a Temporal Action Logic. Journal of Applied Logic (Special issue on Logic Based Agent Verification) 5(2007), 214–234 (2007)
14. Henriksen, J.G., Thiagarajan, P.S.: A product Version of Dynamic Linear Time Temporal Logic. In: Mazurkiewicz, A., Winkowski, J. (eds.) CONCUR 1997. LNCS, vol. 1243, pp. 45–58. Springer, Heidelberg (1997)
15. Henriksen, J.G., Thiagarajan, P.S.: Dynamic Linear Time Temporal Logic. Annals of Pure and Applied logic 96(1-3), 187–207 (1999)
16. Lomuscio, A., Qu, H., Solanki, M.: Towards verifying compliance in agent-based web service compositions. In: AAMAS 2008, pp. 265–272 (2008)
17. Rajamani, S.K., Rehof, J.: Conformance checking for models of asynchronous message passing software. In: Brinksma, E., Larsen, K.G. (eds.) CAV 2002. LNCS, vol. 2404, pp. 166–179. Springer, Heidelberg (2002)
18. Reiter, R.: Knowledge in Action. MIT Press, Cambridge (2001)
19. Reiter, R.: The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. In: Lifschitz, V. (ed.) Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy, pp. 359–380. Academic Press, London (1991)
20. Singh, M.P.: A social semantics for Agent Communication Languages. In: Dignum, F.P.M., Greaves, M. (eds.) Issues in Agent Communication. LNCS, vol. 1916, pp. 31–45. Springer, Heidelberg (2000)
21. van der Aalst, W.M.P., Pesic, M.: DecSerFlow: Towards a Truly Declarative Service Flow Language. In: Bravetti, M., Núñez, M., Zavattaro, G. (eds.) WS-FM 2006. LNCS, vol. 4184, pp. 1–23. Springer, Heidelberg (2006)
22. Yolum, P., Singh, M.P.: Flexible Protocol Specification and Execution: Applying Event Calculus Planning using Commitments. In: AAMAS 2002, Bologna, Italy, pp. 527–534 (2002)

# Towards an Ontological Foundation for Services Science: the Legal Perspective

Roberta Ferrario*, Nicola Guarino*, and Meritxell Fernández Barrera°

*ISTC-CNR, Laboratory for Applied Ontology
Via alla Cascata 56C, 38100 Trento
{ferrario, guarino}@loa-cnr.it

°Law Department
European University Institute
Via Boccaccio 121 , I-50133 Florence
Meritxell.Fernandez@EUI.eu

**Abstract.** As a growing number of economic transactions tend to happen in the Web, their legal implications and assumptions need to be made explicit in the proper way, in order to facilitate interoperability across different normative systems, encourage transparency towards the end users and ultimately promote trust in automated services. In particular, potentially ambiguous terms (and often apparently unproblematic ones) mentioned in these transactions need to be carefully analyzed in order to clarify the distinctions between slightly different meanings, describing hidden relationships and implicit constraints. One of these terms, highly overloaded nowadays, is "service". Indeed, the very fact that services are now offered through the Web, and that the notion of service is at the core of a wholly new organizational paradigm – service-oriented systems – suggests the need to carefully (re)consider this notion. In this paper we shall attempt this analysis under the perspective of *formal ontology,* with a special attention to the legal aspects. The approach we take is that services are complex temporal entities (events) based on the central notion of *commitment.* Analyzing services as complex events allows us to clarify the relationships between the various agents that participate to these events playing specific roles, with specific responsibilities; moreover, this analysis explains a classic economic (and legal) distinction between services and goods, based on the fact that goods are both *transactable* and *transferable*, while services are transactable but not transferable. Assuming that transferability is intended as transferability of *ownership*, we argue that the ontological reason why services are not transferable is exactly because they are events: you cannot own an event, since if owning implies being in control of temporal behaviour, then, strictly speaking (at the token level), the temporal behaviour of an event is already determined, and changing it would result in a different event. So events are not transferable simply because they are not "ownable". Since services are events, they are not transferable as well. Of course, this implies a shared understanding of what ownership, responsibility, duty, right etc. mean, and the paper is a first effort in this direction.

**Keywords.** Ontology, Services Science, Social Service, Service Content, Service Process, Service Description, e-Government, Responsibility.

## 1 Introduction and state of the art

Despite the ubiquity of the notion of service and the recent proposals for a unified *Services Science* [1], multiple inconsistencies between definitions of service from different disciplines (and even within the same discipline) still exist ([2], [3]). In particular, despite the general goal of this science is –arguably- to allow people and computers to smoothly interact with services in the real life, many modelling approaches (especially those focusing on *Web* services) seem to focus mainly on the aspects related to *data and control flow*, considering services as *black boxes* whose main characteristic is to interoperate in a well-specified way (see, for instance, [4], [5], [6], [7]). This black box model has certainly its own advantages, but, according to a recent paper by Petrie and Bussler [8], apparently it seems to work well only within *service parks*, where run-time interoperability is technically feasible because services are very constrained. As the authors put it, "some interoperability among service parks might emerge, but it could take a long time".

Focusing on services as business processes, on the other hand, has its own problems. Overall, the limits of the two approaches (Web services vs. business processes) are well described in a recent note by Katia Sycara [9], who observed that, on one hand,

"*current Web services proposals don't enable the semantic representation of business relations, contracts, or business rules in a machine-understandable way*",

while, on the other hand,

*"current business-process languages [...] are at a low abstraction level and don't provide formal business semantics". In conclusion, "a need exists to model informal business requirements in ways that make it feasible to translate them into precise business-service specifications, including operational interfaces and rules for procedures, timing, integrity, and quality. Such modelling must be driven from the top down, directly from business requirements [...]. The modelling would provide a functionality that's entirely understandable from a business perspective; it would depend on business context, goals, and operational standards, but shouldn't depend on the technology used to implement them. The models would provide business value directly relating to business purposes and could be understood and used without knowledge of underlying IT artefacts".*

This is exactly the perspective we are adopting in this paper, which calls for a broad, interdisciplinary effort such as that envisioned by services science [1]. Under this perspective, we are convinced that a proper, general ontological foundation for the notion of service is a fundamental requirement for such endeavour. This is the long term goal of our work.

The present paper has two main purposes: first of all, we want to explore the foundations of a new ontology of services aiming at establishing a common, unifying framework for representing services according to different views, based on a vision that considers services as complex systems of commitments and activities, involving real people, organizations, and actual circumstances. In other words, we believe it is crucial to take into account the whole *service system* [2] that interacts with Web services through complex chains involving people and computers, which however have always *people* at their ends. That's why in this paper – while trying to be general enough to account for any kind of service – we mostly emphasize the role of *social* and *business-oriented* services, adopting a *global view* which, in a sense, goes against the strict separation between the external and the internal view advocated by semantic Web services standards such as WSMO [10], [11].

The second goal of the paper is that of understanding how *responsibility* is distributed among different agents playing different roles in the whole service system. In order to do so, it is necessary to perform a comprehensive analysis of the notions connected to responsibility, including the juridical implications, which become particularly important in cases where services do not meet the customers' expectations.

A first reason for a global, *transparent box* approach to service modelling comes from the observation that the *terminology* needed to properly expose, retrieve and interact with a service, and especially that needed to understand and negotiate Service Level Agreements (SLAs), unavoidably requires a common understanding of the general service process structure, and the related activities involving the value exchange process between the producer and the customer (see [12] and [13]). Of course, in some cases service producers may have very good reasons for not exposing their internal workflow details, but the point is that, in general, SLAs *may* refer to some details concerning the *way* the service is implemented, whose nature is not specified in advance. So, since the boundaries between the external and the internal service description cannot be defined in advance for all kinds of service, a global approach seems to be the only viable alternative for a foundational ontology of services.

A further reason for a global approach focused on responsibilities lies in the fact that, in many cases, it is important to account for the way a service-based architecture impacts the organizational structure (indeed, service process re-engineering typically impacts organizational re-engineering). In this case, it is crucial to model in the proper way the links between services, people and organizations, where responsibilities play a crucial role.

For sure, modelling services according to this global view is not an easy task, however. The notion of service is so subtle and ambiguous that many researchers simply have given up adopting a clear definition, relying on a variety of intuitive notions mainly coming from practical considerations, which lack unfortunately a coherent framework. In other words, we are still facing the general question: *what is a service?* Is there a single notion behind this term? And if there are multiple aspects, how are they related? How is the internal view of services as *business processes* related to the external view of services as exposed (aggregates of) functionalities?

In this paper we shall address these questions by introducing a novel, general approach to service modelling founded on the basic principles of ontological analysis, centred on the notion of *service commitment* as a temporal state resulting from an agent's promise to guarantee the execution of certain actions in the interest of potential beneficiaries in correspondence of certain triggering events. In this view, services are modelled by means of a layered set of interrelated temporal activities, each one with its own participants and spatiotemporal location. Under this respect this approach shares many similarities, in its main inspirations, with Alter's work on service systems [2], as well as O'Sullivan's work on non-functional requirements for services [14], and Baida, Gordijn and Akkerman's work on the service value chain [15]. This proposal is meant to be a first concrete step towards a unified, rigorous and principled ontology. The analysis of the service system's structure and of the responsibilities distribution across this structure is necessary in order to build a model which is as faithful as possible to the social, business and legal perspectives.

The legal perspective, which is the main focus of this paper, is especially important in a world where international economic and political interaction requires communication between different legal systems, and ensuring mutual understanding becomes a strategic goal for guaranteeing conflict prevention and resolution. Indeed, the lack of unified conceptualizations in legal discourse can flaw the whole communication process, giving place to legal uncertainty, diverging interpretations of legal relations and – consequently – impasse situations that can only be solved through costly judicial procedures. Since the notion of service is a central one both for private and public law, the development of a foundational ontology of services plays an important role in the construction of a conceptual language for legal analysis. This paper aims therefore at providing a framework for the design of an abstract service model potentially applicable to different legal systems, where the notions of responsibility, liability and delegation are structured together, thus enabling the analytic description of services together with their juridical implications. Such detailed description, based on general legal notions such as duty, right and obligation, offers a formal framework useful to assess real life interactions and draw conclusions on the responsibility and liability of different actors. This is especially relevant nowadays where the puzzling complexity of contractual relations makes the task of liability allocation particularly intricate.

## 2 The proposed approach

### 2.1 The Basic Idea

If we start from the simple question "what is a service?", it is immediately very evident that there is a huge confusion, not only in the layman's common sense, but also in the way the term is used in the literature. Sometimes the term "service" is used to indicate an *action* (actually performed by somebody), or a generic *type of action* (including in this category data manipulation procedures such as those typically described as Web services), or perhaps the *capability* to perform some action; other times it refers to the *result* of such action, which is typically a *change* affecting an object or a person, or just the (subjective) *value*, or utility of such change; moreover, in certain settings (like Public Administrations) the term often denotes an *organization* acting (or in charge of acting) in a certain way in the interest of somebody.

In our opinion, all these notions are somehow connected, and contribute to better specify the notion of service, but none of them can be properly identified with what we believe people are commonly referring to when asking for a service. More or less "official" definitions occurring in the ICT literature do not help much (some relevant exceptions are [2], [14], [15] and [16]).

To see how these various definitions are related, let us start with some simple questions, focusing on very general public services such as fire-and-rescue, snow removal, children care, etc. What do we *pay for*, when we fund such services with our taxes? What does it mean that, for instance, in a municipality there are such services, at a certain time? Is anybody extinguishing a fire or removing some snow *at that time*? No, certainly not necessarily. We can legitimately say that *here and now* both a fire-and-rescue service and a snow removal service are *present*, even though there are no lit fires, nor is it snowing. It suffices to say that there is someone (firemen, snow removal operators) who is *prepared* to perform precise actions in case something happens (fire, snow). So our core notion of service is based on the following statement:

*A service is present at a time t and location l iff, at time t, an agent is explicitly committed to guarantee the execution of some type of action at location l, on the occurrence of a certain triggering event, in the interest of another agent and upon prior agreement, in a certain way.*

So, in a sense, at the core of any service there is a *commitment* situation in which someone (the service *provider*) guarantees the execution of some kind of *action(s)*[1] in the interest of

---

[1] Alternatively, instead of focusing on *actions* to be executed, one can decide to treat commitment as directed towards *conditions* to be achieved, as it happens in [1]. In this case, the commitment's content can be expressed by a proposition. In the perspective of services, this difference amounts to focusing on the actions that must be executed in order to reach a certain desired state or directly on the achievement of such desired state. We could in fact leave this choice open, as the two alternatives seem both very reasonable: sometimes the customer is just interested in some particular state to hold, disregarding how it has been reached, other times the commitment concerns the action to be performed, even independently of the actual achievement of a specific result. In any case, we believe the latter case is more frequent and thus representing a commitment with respect to actions is more in line with our analysis.

somebody who agrees (the *service customer*), at a certain cost and in a certain way. This action is executed by the a *service producer*, who may coincide with the service provider, may be somebody else *delegated* by the service provider, or even coincide with the service customer (e.g. in rental services, where the action of using the rented good is actually performed by the service customer).

From the ontological point of view, this commitment situation is a static temporal entity, i.e. a *static event* in the sense of the DOLCE ontology[2] [17], which involves the participation of a single agent, the *provider*. This commitment state typically starts at the time of the commitment act, and its duration is determined by the commitment's act itself[3], which typically specifies some constraints concerning the way the commitment will be fulfilled.

As we shall argue in the rest of the paper, service commitment needs to be distinguished from *service content*, which concerns the kind of action(s) the provider commits to guarantee, and *service process*, which is a set of business processes implementing the service commitment (see Figure 1). In turn, we distinguish service commitment from *service availability*, which involves a service process running at a certain time and location: this allows us to account for malfunctioning periods or working pauses, where the commitment still holds but the service is not available. Following [18], [19], [20] and [21], the commitment act can be seen as a *speech act* that most of the times is codified in a *document*, i.e. in an institutional object that can assume many different forms: a contract, an official declaration or deliberation, a service level agreement[4], etc.

In institutional settings, the *provider*, the agent who commits, is typically a Public Administration. On the other hand, the service *producer*, who actually executes the action(s) guaranteed by the provider, may not necessarily coincide with the provider, and can be either a PA or another kind of (private) organization, delegated by the provider; in some exceptional cases even an individual agent. The same holds for the *service customer*, who can in turn be a PA, an organization, or an individual agent, the latter being much more common than in the previous case.[5]

The last element present in the definition is the *triggering event*; two kinds of triggering events can be singled out. The first one, more trivial, is a simple request made directly by the customer (like a parent in need who requires children care); in this case the *service invocation* coincides with the triggering event. The second one is the occurrence of a particular event kind, like the lighting of a fire in a wood, or a difficult situation observed by a social assistant, that triggers the action[6]. Of course, since the occurrence of the triggering event is not known in advance, the action time is in general much shorter than the availability time, so a service may be available at a certain time even if none of its foreseen actions do actually occur.

---

[2] Although the term "event" has often a dynamic connotation, we use such term in the more general sense of *entity which occurs in time* (also called *perdurant* in the DOLCE ontology). In this understanding, states and processes are considered as special event kinds.

[3] We assume that the commitment act (the *speech act*) is instantaneous, and occurs at a time which does not necessarily coincide with the beginning of the availability state.

[4] In the actual practice, the term "service level agreement" typically refers to the negotiation that the producer conducts with the user; here we are using the locution in a coarser sense, which includes also the provider-producer and provider-user agreements, as well as, possibly, those between the provider and the community to which services are provided.

[5] In some cases, like in rental services, the service customer may coincide with the service producer. We do not discuss this case here, however.

[6] To be more precise, it is the *observation* of such event that triggers the action. It is worth noting that, for this reason, many services include among their supporting activities an explicit monitoring activity, which can be executed by the producer itself or delegated to another agent.

It is worth stressing an original feature of our definition, namely the inclusion of the triggering event. Traditionally, approaches on services are goal oriented; take for instance the definition from [22]: “A service delivers a process to achieve a certain goal by using resources”. Note however that actually the goals may in some cases be just implicit, or even different if you compare the producer’s perspective with the customer’s perspective. In such cases, specifying the service also in terms of the triggering event and the action to be performed in correspondence of such event seems to be less ambiguous. The service’s goal doesn’t disappear in our approach, and indeed it is present in what has been called the service content, but the triggering event allows to justify the passage from service availability to service invocation. Moreover, note that a triggered action may not necessarily succeed. What the provider guarantees may in some cases be only the action’s performance, not its result. This changes also the mechanisms for the evaluation of service quality, which must distinguish between actions/processes and resulting states.

### 2.2 Services and goods

To better understand the nature of our proposal – that services are temporal entities (events) based on *commitments* – let us briefly discuss the difference between services and goods. According to the World Trade Organization, services are a sort of intangible goods, so that a service might be defined as anything you can buy, but “you can’t drop on your foot”. Yet, Ted Hill [23] insists on the fact that services are not a special kind of goods, because goods and services belong to quite different ontological categories: goods are both *transactable* and *transferable*, while services are transactable, but not transferable. In Hill’s own words, “a surgical operation is not some kind of immaterial drug”: when you buy the drug you become an *owner* of it, in the sense that you can decide about its behaviour (i.e., assuming it in your body), while when you pay for the surgical operation you are not actually becoming the owner of it[7]. In support to this argument, we argue that the ontological reason why services are not transferable is exactly because they are events: you cannot *own* an event, since if owning implies being in control of temporal behaviour, then, strictly speaking (at the token level), the temporal behaviour of an event is already determined, and changing it would result in a different event. So events are not transferable simply because they are not “ownable”. Since services are events, they are not transferable as well.

So, in conclusion, it seems legitimate to assume that goods are *objects* (endurants, in DOLCE’s terms), while services are *events* (perdurants). One may observe however that our economy is full of examples of transactions involving services, where the service seems to pass from hand to hand: certainly somebody may buy Amazon, for instance: our point is that in this case the transaction involves the Amazon *company*, not Amazon’s *service*: there is a change of ownership concerning the service producer, but not the service itself, which remains the same (as long as the Amazon company doesn’t change its legal identity, and its service content – the actions it offers – remains the same).

***Transferability of services as transferability of a right***

We have seen that a service, being an event, is not transferable. However, certainly we can transfer a right *to a service*. In the context of services, a right to a service implies a deontic position, in the sense that someone has a right if there exists a certain corresponding duty. A right of *A* of receiving a service would correspond then to the duty of *B* to provide the service, namely, to the duty of participating (as the main agent) in the *commitment event* that is

---

[7] An interesting analysis of the notion of ownership can be found in [24].

at the basis of every service. A right can be reified, that is, considered as an object, and this is actually a common move in the legal domain[8]. And a right can be *owned*, in the sense that somebody can decide how to affect its "behaviour" (e.g., deciding *when* the right is claimed).

What is the conclusion of this conceptual analysis with regard to the notion of service? That even if a service is not directly transferable, since it is an event, *rights to services* are transferable: the object of the transmission is in this case the right, and the service is the event which the right refers to. In other words, having the right to a service is the same as being able to legitimately claim the performance of a certain service (set of actions) by some other party. And what can be transferred, from a legal perspective, is precisely this legal position that enables someone to legitimately claim the performance of a service.

A similar analysis can be applied to the notion of "good". In this case however the ownership (intended as the capability to influence the temporal behaviour) concerns the good itself, and not just the right to use it. In other words, we can own a good *in addition to* owning the right to exploit it, while for services the only thing we can own is the right to exploit them. Note that, although in many cases ownership transfer for goods implies a physical transfer (change in physical location), this is not always so, as in the case of real estates.

***Transferability of services as transfer of a duty***

From the provider's point of view, a service transfer might be understood as well as the *transmission of the duty to provide the service content*. However, we should distinguish between *total* transfers of this duty from an agent to another (including all related responsibilities towards the service customer), and *partial* transfers (i.e., *delegations*), which maintain some responsibilities on the side of the delegating agent. The latter case is typical of public services, whose content is to be guaranteed by the state. Indeed, the public entity can *transfer a part of the duties* that compose its obligation to a private entity –namely, the obligation to actually *produce* the services in specific circumstances- but not its general duty and it will therefore keep part of the responsibility. Let us imagine health services. Health service can be a public service in the sense that the State has the responsibility of guaranteeing it. The State can, however, agree with a private party (private medical company) that whenever citizens need health assistance, the private party will provide it. The responsibility over the service remains in the State, since it cannot completely delegate this duty[9]. However, the particular duty of satisfying concrete needs arising from specific situations (e.g. concrete medical care needs derived from the situation in which someone breaks his leg) is assigned to the private entity to which the service is entrusted. The duties related to health services are thus split between the state (the *provider*) and a private actor (the *producer*). A similar situation occurs in the case of subcontracting, by which the general contractor delegates to the subcontractor the performance of a specific task, keeping nonetheless responsibility towards the client.

On the contrary, a *total* transfer of duty can happen in the private sector, where we can distinguish two cases, depending whether the duty comes from a *generic* commitment towards potential customers, or a specific commitment concerning an *actual* customer, under specific circumstances. In the first case, it is common to talk of business transfers. This means that, if a business provides a private service with no special limitations on its transfer (and assuming that the service buyer fulfils all the necessary legal requirements), the full responsibility is simply transmitted to the acquiring party (let us imagine for instance the previously mentioned transfer of Amazon, or the transfer of a restaurant, or a hairdressing salon). In the sec-

[8] Rights were already clearly distinguished in Gaius' classification of the law: "Res incorporales: things which cannot be touched, such as those consisting in rights, e.g. an inheritance, a usufruct, obligations".

[9] And this would therefore imply that complaints can still be directed to the State in case the service provided by the private company does not work well.

ond case, the transmission of a particular service, that is, of the concrete obligation to provide a service to a party, will be possible if this does not substantially alter the quality of the service contracted by the party or, if otherwise, the latter provides consent (let us imagine for instance that we reserve a room in a hotel and when we get there the hotel is full and they send us to another hotel within the same group).

As a conclusion, let us insist that, despite the cases above can be described as service transfers in the everyday jargon, properly speaking, according to our definition, the transfer does not involve the service itself, but rather certain *normative positions* (duties or obligations) that refer to it. Moreover, we maintain that, while an internal delegation process concerning the actual *production* of service content certainly does not alter the identity of a service under our definition, a *total* transfer of duty implies the existence (or the creation) of *another* service, different from the original one. If the owner of Amazon changes, while Amazon maintains its legal status and its service content remains the same, then the service remains the same, because the provider is still the same. On the contrary, if the provider changes (e.g., because the company dissolves after being bought by another), then the service changes, although the service content may remain the same. Suppose that for instance there are two companies, A and B, providing exactly the same service content (possibly at different prices). According to our definition, we say that there are two distinct services. If now A buys B (that is, the *company* B, with all its duties and rights, is transferred to A) the result is that the service provided by B disappears, and only one of the previous services survives.

### 2.3 The basic ontological structure of services

Let us continue our analysis with another question: what's happening when a service is *produced*? As we have seen, a service may be concretely available even if it is not actually delivered, or maybe will be never delivered: we keep paying the firemen even if no fires occur. So, in our approach, a service has to be distinguished from its actual *delivery* to a particular customer. Indeed, typically the same service guarantees multiple deliveries. By the way, to avoid confusions, we propose an important terminological distinction: strictly speaking, it is not *the service* which is delivered, but its *content,* i.e., the actions intended to be performed in the interest of the customer. So a service implies first of all a concrete commitment (from the side of a *provider*) to guarantee the production of a certain content, consisting in actions of a certain kind executed in a certain way. Altogether, the various actions that ultimately lead to service content production (performed by the service *producer* on behalf of the provider) constitute the *service process*. We shall say that a service process *implements* a service commitment. The concrete delivery to a particular customer presupposes however a *service acquisition* activity engaged by the latter, which typically negotiates a *service offer* resulting from *service bundling and presentation* activities on the producer's side. Finally, to complete the picture, we have to take into account the activities related to the value exchange chain, which include the service exploitation from the customer's (*customer's exploitation),* the sacrifice needed to access to the service (*customer's sacrifice),* as well as the corresponding activities from the producer's side.

So, as illustrated in Figure 1, a service is conceived as a complex event, with five main proper parts: service commitment, service presentation, service acquisition, service process, and service value exchange. In the following, we shall discuss these notions in more detail, with the aim to establish the basis for an ontology of services able to account both for *service descriptions* from an external point of view (typical of Web services and Service Oriented Architectures) and for *service processes* from an internal, business modelling point of view.

First of all, let us remark that all the blocks described in Figure 1 are *events* (*perdurants*, in DOLCE's terminology). This means that they can be characterized, roughly, by their *tempo-*

*ral location* and by their *participants*, linked to the event by means of what are usually called *thematic relations*: *agent, patient, theme, instrument*… Specifying a service (or a service kind) amounts to constraining these events by imposing suitable restrictions on their temporal locations and thematic relations. So, for instance, non-functional requirements such as those discussed in [14] are represented as attributes of specific service components, each involving a particular aspect (participant/thematic relation) of a particular service event. The resulting analysis, which we cannot discuss in detail for reasons of space, looks very similar to Alter's *service responsibility tables* [2], where the rows correspond to service components (events), and the columns to specific aspects to be considered (thematic relations). We give therefore a clean ontological foundation to a business-oriented proposal. Moreover, specifying the agents involved in each event allows for a fine-grained account of the *organizational impact* of a certain service. Note that, although the relationships between these various events (for instance, whether or not they involve the same agents) may vary according to the nature of the service specified, there exists a systematic ordering relationship between them, so that a service has a *layered structure*. This ordering relationship is not so much a temporal precedence (indeed most of these events are temporally overlapping), but rather an (existential) *ontological dependence* relationship: in order for an event at a certain layer to occur, some event at the higher level has to occur. Ultimately, all the events belonging to the service process presuppose some acquisition event, which in turn presupposes the service commitment.

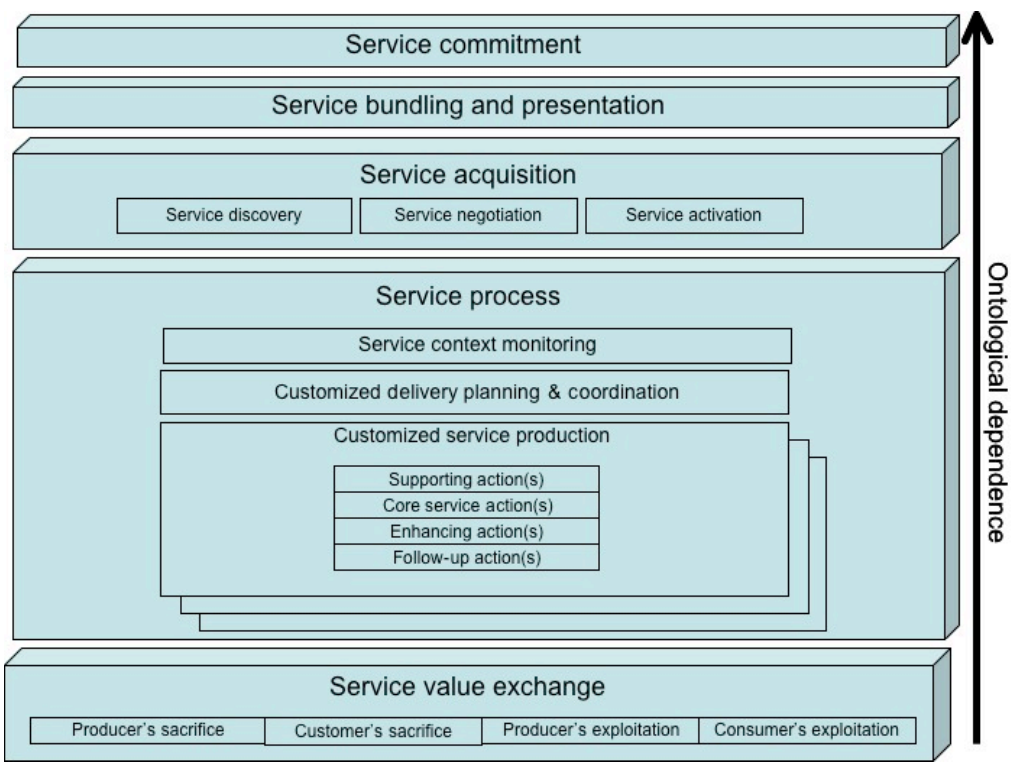


**Figure 1:** The layered structure of service activities

At this point it is important to notice the central role played by actions and events in this account for the description of services; this is in contrast with the major trend, which is to describe services in terms of pre-conditions and post-conditions, like in WSMO, where processes are represented as transitions between states [11]. Under a different perspective, the two approaches could also work in conjunction.

There are several reasons why in our opinion it is important to explicitly represent events; first of all, though for the front-office in most cases it is enough to know which is the starting state and which the desired one, for the back-office it is important that the whole process be transparent (to know who does what and when), especially when a failure is at stake. But malfunctioning is not the only reason: without events one does not have sufficient expressiveness to distinguish two different commitments with the same content but different deadlines. Even if one sees these deadlines as non functional properties, it is hard to use them, say, for expressing a SLA without a clear semantics.

Again, in [25] a list of service quality's determinants is given, in which at least a couple of these determinants are strongly space-time dependent: responsiveness, connected with timeliness of service and access, defined by (among others) three items: "waiting time to receive a service […]; convenient hours of operation; convenient location of service facility."

By using only pre- and post-conditions other subtle but important differences are lost. Take for example these two scenarios: 1. an unemployed woman who becomes pregnant and 2. a pregnant woman who becomes unemployed. In our account, the two scenarios can be distinguished by the fact that in 1 the pregnancy is the triggering event, while in 2 it is the unemployment. This difference may result in the activation of different services (for instance, a financial aid in 1 and a help in searching a new job in 2, or a legal enquiry on the employer if there's a suspicion of unfairness). In a pre- post-conditions framework both scenarios have the same pre-conditions and thus should activate the same services. Notably, the literature in economics has since long recognized that comparing the outcomes of services is not enough in order to evaluate their quality. See for instance [25] (similar distinctions appear in [26], [27], and [28]):

> Quality evaluations are not made solely on the outcome of a service; they also involve evaluations of the *process* of service delivery.

Let us now consider the various events constituting the service process internal structure. In Fig. 1, the containment relationship between the various blocks represents the parthood relationship. The core constituent of a service process is a set of basic activities (each called *customized service production*[10]), centred on the delivery of service content to a *single customer.* In addition to the *core service action(s)* depending on the service nature, a customized service production may include *enhancing actions* intended to increase the service value or differentiate it from those of competitors [23] as well as *supporting actions* needed to enable the core service consumption and *follow-up* actions intended to monitor the core action's results. In addition to customized delivery activities, the service process includes various back-office activities concerning *customized delivery planning and coordination*, plus an activity we have labelled as *service context monitoring* –which seems to be neglected by most current approaches – which involves the various actions necessary to detect the event that triggers service production, which can be an external situation or a customer's request: without an explicit modelling of such activity, there would be no way to account for delays or improper management of triggering events.

As a presupposition to service production, typically some *service acquisition* activities are required from the side of the customer[11]. These include *service discovery*, which is the event

[10] In the context of public services, a single event of customized service production is often called an *intervention*.

[11] Even in the case of free, public services, it is difficult to imagine a case where the customer is not required to actually discover the service, or make a minimal sacrifice to exploit it.

where the service provider (or producer) and the service customer first meet together; *service negotiation,* which involves an agreement between the two parties, and *service invocation,* which refers to the event where the customer agrees to the service (not necessarily implying immediate production).

On the other hand, the service production results in a complex chain of transfers of value, which are represented in Figure 1 as the event Service value exchange. With a simplification, this is decomposed in *Producer's cost*, *Customer's cost*, *Producer's revenue* and *Customer's revenue*.

While in the case of the producer, most of the times (both for cost and for revenue) the value has to be intended in terms of money, in the customer's case things are more complicated. For instance, especially for services in the social domain, the customer's cost can be seen as an action whose results go somehow against some of the recipient's desires, but which the customer is still willing to perform, like the *service sacrifice* mentioned in [11]. Also the customer's revenue sometimes is not expressible in monetary terms, but only as some wellness state. Moreover, even though there's always an ultimate recipient of a service, we could also have indirect recipients, like the community that pays with its taxes for the service and benefits in terms of enhancement of its social conditions.

In conclusion, we can say that a service is characterized in a *prescriptive* way (*commitment level)*, while a service process in a *descriptive way* (*implementation level)*. The commitment level is where the "rules of the game" are established: what types of action compose the service, what types of agents are entitled to execute those actions, what types of agents may qualify as recipients, what types of events can become triggering situations. It is also the level where legal responsibility is at stake. In fact, from the point of view of the service offering, it is not important who in particular executes certain actions, but rather that a certain kind of action is executed in a certain way, by an agent who displays certain features and has some competences. The agent who is responsible that the required conditions are met is usually an organization, such as for instance a public administration. Such responsibility is usually distributed and assigned according to some structural constraints, i.e. by devising a structure of roles and sub-organizations. The ontological analysis of organizations is thus a topic tightly connected to the ontological analysis of services.

When we come to the actual service process, the various *kinds* mentioned at the commitment level need to be instantiated in concrete *tokens*. Individual agents are those who realize the core actions of service production, whose recipients are, ultimately, concrete agents (citizens); also the triggering situation is the occurrence of a precise (instance of) event. The service production level is thus characterized at the *descriptive* level, the one the data that are recorded and transferred belong to.

Finally, let us mention the issue of spatio-temporal location of services. In very general terms, one could say that in most cases when a somebody makes available a service, this availability spans over a spatio-temporal region which includes the spatio-temporal region in which the core service actions will (possibly) be executed; in rare cases, the two can coincide. For some special services, the analysis can be further complicated by the fact that the service may be delivered in a place and at a time and received in another place at another time. We won't enter into these details at present, but the issue needs to be investigated.

## 3. Responsibility, right, duty, obligation

### 3.1 Connections between the main notions

Before exploring in more detail the juridical implications of services in terms of the layered structure of interrelated events described above, let us focus in this subsection on the notion of responsibility, which is quite central in the study of services, analysing it under many respects, including its significance from a juridical point of view and its connection with the related notions of right, duty and obligation.

For a start, a definition of responsibility that can be found on a common business dictionary[12] is the following: “Duty or obligation to satisfactorily perform or complete a task (assigned by someone, or created by one's own promise or circumstances) that one must fulfil, and which has a consequent penalty for failure.”

As we can see, this definition includes both the possible sources and the common consequences of responsibility. As possible sources it mentions both an exogenous assignment (as in the case of delegation, that we will see in a while) and a promise coming from the same agent who’s taking the responsibility (as in the case of commitment). Note that, if there is an exogenous assignment, the corresponding commitment is not necessary: for instance, a government may be responsible for the population’s health without actually committing to it. In any case, the consequence of having (undertaken) a responsibility is to become the bearer of an obligation that, if not fulfilled, would bring about a sanction (which is the last element of the definition above).

When responsibilities are institutionally established and codified in a contract (such as a Service Level Agreement), then a relationship originates between the service provider and the service customer, such that then each customer’s rights is connected to a correspondent obligation for the service provider; very often this obligation is accompanied by a possible sanction, in case the obligation is not fulfilled.

The right of the customer to have the service fulfilled, and the belief that the agent will be sanctioned in case it is not, engenders an expectation in the customer that the service will be delivered.

If we look at it from a juridical perspective, *responsibility* refers to the situation of being accountable to someone for something. There exist many types of responsibility depending on the kind of normative order that makes one accountable: social responsibility if the normative order is the social order, moral responsibility if the normative order is a particular moral system and legal responsibility if the order is a legal system, among others. The core meaning of the notion of responsibility involves therefore the existence of a certain kind of normative order that should be observed.

In the legal domain, a difference is made between the general notion of social responsibility and a stricter notion of *liability*, which refers to the elements that are generally required for incurring legal blame because of the violation of a social responsibility, and, in private law, for being compelled to comply with a court order to pay damages or compensate in some other way the damage done [29]. Criminal liability entails punishment, which frequently takes the form of imprisonment. This way, legal responsibility (liability) can be seen as the concept that enables to blame someone for unlawful action. Once the responsible subject is declared liable, actions can be taken to compel him or her to compensate the harm done or to punish him or her.

Legal liability requires the existence of an agent, whose conduct is considered legally relevant; a patient, affected by that conduct; the conduct or action; and the wrong or harm, understood as the effects on the patient. Agent and patient do not necessarily have to be concrete physical persons; in private law both agents are private parties, in public law the harm is in general terms not merely caused to an individual party, but to the community or to the general interest (this is one of the criteria for distinguishing private law and criminal law).

Note that, as observed above, legal responsibility (liability) inherently requires *a pre-existing normative order against which to judge the agent's conduct.* This normative order

---

[12] http://www.businessdictionary.com/

can be translated into particular *normative positions* (rights and duties). These normative positions define the liability-responsibility relation and permit to allocate specific roles to the agents involved. A landmark contribution to the notion of duty and right is [30], where an analysis of fundamental juridical notions is presented. In this work "duty" and "right" are considered correlative concepts that can be represented through the following scheme: *"[...] if X has a right against Y that he shall stay off the former's land, the correlative (and equivalent) is that Y is under a duty toward X to stay off the place.*" [30]. If X's duty (or obligation) is not fulfilled, then there will possibly be legal responsibility and liability.

Taking into account this general framework and according to the model of services presented above, in the context of services provision we can envisage different sources for legal responsibility (liability):

- breach of contract (of obligation) [*contractual responsibility*]
- specific damages resulting from the breach of contract [*contractual responsibility*]
- specific damages resulting form the wrongful performance of service: civil and criminal responsibility [*extra contractual responsibility*].

These different types of legal responsibility (liability) are connected to the breach of different kinds of duties or obligations. Let us imagine a public service, namely, health care. We can distinguish different duties in this regard:

1. the duty of providing particular health care services (doctor assistance, medical tests, diagnosis, treatment, ...) at particular moments and to particular individuals. It is therefore a contractual obligation or duty.
2. the duty of performing the actions directed to provide health care services according to general diligence (standards of conduct in the medical field determined by the community of medical professionals and their practices) and according to certain conditions explicitly established in a certain context. These latter conditions can be established by the regulation (in a particular country) for health service providers. It could be regarded therefore as an extra-contractual obligation or duty.

If duty 1 is not fulfilled, this can be seen as a breach of contract (obligation) and as specific damages resulting from the breach of contract; whereas if duty 2 is not fulfilled, this could be regarded as damages resulting form the wrongful performance of the service: civil and criminal responsibility.

The previous scheme can be applied to the framework of public services. The cases presented can be understood in terms of abnormal function of the public service, according to Principle I of the Council of Europe Recommendation R (84) 15 of the Committee of Ministers to Members States relating to public liability:

> *"Reparation should be ensured for damage caused by an act due to a failure of a public authority to conduct itself in a way which can reasonably be expected from it in law in relation to the injured person. Such a failure is presumed in case of transgression of an established legal rule".*

However, state liability is not excluded in case the function of the service is normal, if it is considered unfair that the citizen bears the damage. This is stated in principle II.1:

*"Even if the conditions stated in Principle I are not met, reparation should be ensured if it would be manifestly unjust to allow the injured person alone to bear the damage, having regard to the following circumstances: the act is in the general interest, only one person or a limited number of persons have suffered the damage and the act was exceptional or the damage was an exceptional result of the act."*

The transferability of these different responsibilities varies. Responsibility travels along with the corresponding normative position, that is, with the specific duty. If the duty is transmitted to a private party, the corresponding liability is as well transmitted. Nevertheless, the more abstract statutory duty consisting in the obligation to guaranteeing the existence of a structure or organisation that provides a particular service is not transferable. This duty is established by laws, quite often constitutional texts, which determine that the state is responsible of guaranteeing certain public services to which citizens have a right. This is why some peculiarities exist in the regime of public responsibility. For instance, the establishment of strict liability, which affects not only the state, but as well the private bodies that are providing the service on the basis of a license [31]; or, even if the service (duty 1, duty 2) has been transferred to a private entity, complaints might have to be directed to the state (this was once the case in Spanish regulation [32]). The state could be held liable as well, for instance, in case the statutory certification bodies had not performed well their task and had granted permission to operate to a private medical centre that did not fulfil the minimal conditions and some concrete injury had been caused to a patient. It could be held responsible as well in case the damage was due to a compulsory clause imposed by the state in the public contract granting permission to the private entity to provide the service.

### 3.2 Patterns of responsibilities, obligations and rights across service structures

The starting point of our analysis is the event of service commitment, in which a service provider commits with someone (a community, or an authority) that a certain service content will be produced for the benefit of a designated kind of customer.

In the literature a distinction is traced between implicit and explicit (or "explicitly represented", in Singh's terms) commitment, that goes back to the more traditional approaches, like that of Becker [33], who distinguishes between commitment by default and commitment by conscious decision. In institutional settings the commitment is usually codified in a document, like a contract (and gives rise to contractual responsibility); this contract creates new juridical entities: it creates an obligation for the provider; this may be a direct obligation to perform certain kinds of actions or it can consist in seeing to it that such kinds of actions are performed by someone else. In this latter case the contract envisions also a delegation action. When a provider delegates the execution of a service to a producer, the action creates an obligation on the side of the producer (of executing the service) and a right from the side of the provider (of having the service executed, so as to be able to fulfil its previously determined obligations towards the community or authority whom the commitment was addressed to). Delegation relations may be seen as responsibility transfers between agents. As noted in the previous subsection, if the duty of executing the action is transferred (delegated), also the responsibility of that action is transferred; nonetheless, the obligation of guaranteeing that the service is executed (which pertains the public authority that plays the role of provider) is not transferable. It is interesting to notice that in [21] Singh explicitly lists delegation among the operations that can be executed on commitments. He also highlights the fact that when a commitment is delegated, agents shift their roles: he says that the role of debtor is shifted, that is to say that the commitment passes from the provider to the producer, who has now the responsibility with respect to the execution of the service actions. We are aware of the fact that this description is a simplified one, as the responsibility can sometimes be shared in varying proportions.

In [34] a classification of different types of delegation relations based on three dimensions is traced. In particular, along the first dimension, based on the nature of interaction, the authors distinguish between weak, mild and strong delegation: in weak delegation there is no agreement, no requests and no intended influence, so that the one who delegates just exploits the actions of the other; in mild delegation there is still no agreement and no request, but the desired behaviour is in some way induced; finally, in strong delegation there is an explicit agreement on which the delegation is founded. It is obviously this latter type of delegation that is at stake between service provider and producer, as the delegation in this case comes into being just with the signature of a contract or some other formal agreement.

Another dimension that is of interest for this discussion is the one based on the degree of task specification, that distinguishes between open, close and intermediate delegation, depending on whether the object of delegation is more or less specified, in a spectrum ranging from open delegation with minimal specification to close delegation with complete specification, with various degrees classifiable as intermediate delegation.

The degree of specification of the delegation depends on what is written in the delegation agreement or contract, similarly to the case of the service commitment contract, where the commitment's content can be more or less specified concerning the way the core actions are executed.

Disregarding who is in charge of executing the service actions, the commitment also creates a right on the side of the customer.

Even though, as we just mentioned, in most cases public institutions make their commitment public through a contract or a deliberation, it happens very often that, while the assumption of the commitment on the side of a public entity is explicit, its acceptance by the beneficiaries is only implicit and given "by default" by their belonging to a particular social community. This default acceptance assumption ensures that the beneficiary can claim the execution of a certain service even without having signed any contract or having negotiated anything with anybody on this respect.

The chain of rights and obligations generated by the commitment event also imposes various kinds of constraints on the other events composing the service. For example, during the activities of service bundling and presentation, the service producer is constrained in that it cannot promise anything that is excluded by the contract signed at the time of the commitment event. Also, what is advertised in the presentation must not be anything that cannot then be executed. The fact that the service is presented in a certain way may also give additional rights to the customer, like the right to have it executed in the way in which it has been presented (not always, sometimes if the contract doesn't explicitly commit to execute the service as it is advertised in the presentation phase, it may be that the customer does not acquire such right, but at least the producer becomes liable of being sanctioned for what it has promised and then not fulfilled. In this case maybe the customer can be refunded, thus acquiring another, different, right).

Also the service negotiation phase is very important from a juridical standpoint, since when a new contract with a specific customer is signed it has to comply with what established in the general commitment contract, but it can add details to that. This customized contract makes the producer's obligations more precise (possibly adding new obligations) and creates specific rights for individual customers (differently from the commitment, where rights were attributed to classes of customers).

The service process is the phase in which what has been promised in the commitment and in the negotiation phases is realized. According to the level of detail of the contracts resulting from these two phases, the various actions of the service process can be executed in a more or less pre-defined and specific way.

It is important to notice that the service process also includes a service context monitoring activity. The way this monitoring is performed is also regulated; usually, this regulation is primarily given in the commitment, where the provider also commits on monitoring, but can decide whether to execute such monitoring directly or to delegate it to someone else. This delegation can be directed either to the producer of the service, or to a different entity, that is then only in charge of the monitoring. In case it is delegated to the producer, this can further delegate it to some other entity.

The commitment towards monitoring (be it direct commitment or a commitment acquired via delegation) and the consequent rights and obligations that it brings with it, underline the importance of the triggering event in the structure of the service. If the entity in charge of monitoring fails to detect the presence of the triggering event, the service process cannot be initiated. In this case it is not the service producer that has to be deemed responsible for the failure of the service process, since this was not initiated due to the lack of the “start signal” given by the detection of the triggering event.

Finally, after having verified that in the service process phase all that was promised in the previous phases has been properly realized, a right of receiving some income on the side of the producer arises in the service value exchange phase. Consequently, for the provider (or the community in public services’ cases), an obligation of providing such income to the producer is created. The amount of such exchanges is usually fixed in the negotiation phase. If the contract also specifies some constraints in the customer’s costs (for instance, the fact that a service has to be produced in a restricted timeslot) and these are not met, a new negotiation phase can take place.

## 4 A revised version of Alter’s responsibility tables

An author who also deems the concept of responsibility as central for service science is Steven Alter, who, in a recent article [2] has presented a conceptual instrument that he calls “service responsibility tables” (SRT); these are aimed at facilitating a better understanding of services primarily based on the responsibilities assigned to each role; moreover, Alter suggests to add as many columns as necessary in order to address different aspects of analysis. In practice, Alter isolates two orthogonal components of services: the constituting actions and, for each of these actions, the responsibilities of the involved stakeholders; he thus describes *how* such stakeholders participate to the various events. These *modes* of participation individuate the *role* the participants play in the various events constituting the service.

Even though the topics suggested by Alter are heterogeneous and sometimes confusing, we are interested in his idea of representing the events composing a service and the role participants have in these events and in using the tables to represent the distribution of responsibilities across the events composing the complex structure of the service system.

In order to represent all this, we take inspiration from a notion introduced in linguistics to account for the internal structure of events: so-called *thematic relations* (or *thematic roles*), expressing the nature of the relationship between an event and its participants. Adding thematic relations to those linking an event to its own qualities (such as temporal and spatial location) we have a full set of attributes at our disposal, among which the following ones appear to be as especially relevant for our purposes:

- Agent (the active role, the one who acts in the event)

- Theme/Patient (the one who undergoes the event; the patient changes its state, the theme does not)
- Goal (what the event is directed towards – typically a desired state of affairs)
- Recipient/Beneficiary (the one who receives the effects of the event)
- Instrument (something that is used in the performance of the event)
- Location (where the event takes place)
- Time/duration (when the event takes place, or how long it lasts)

As a result, in the service responsibility table we have the main composing events in the rows (service commitment, service acquisition, service process, service value exchange…) and the thematic relations in the columns.

In order to give an idea of the approach, we take an example and we represent it using the tables. The example is directly taken from [23] and it is about a guy who goes to the mechanic's garage to have his car repaired. The aim of Table 1 is that of representing in an explicit way the fundamental constraints that need to be specified in an actual service description. This can bring many advantages both in the comprehension of the service's features and in the many different evaluations of service quality that can be made under various viewpoints, among which the legal one, that is particularly relevant in this context.

The table describes the events in which a generic car repair service is articulated. The values we put in the various cells allow us to express the relevant constraints that distinguish this service from others.

We start with service commitment. During the commitment event, that chronologically comes first and is the one that all the other events depend on, the garage's owner commits with a Public Administration (for instance the Chamber of Commerce) with a subscription act and his commitment consists in guaranteeing that someone (the mechanic) will execute a certain type of job (illustrated in the job description, on which he commits) according to the local rules. This commitment is valid in the whole Province (for instance) and starting from that very moment on.

After the commitment, we have the service acquisition, which in turn is composed by three different events: discovery, negotiation and activation. During discovery the customer looks for a garage (that is then the *theme* of his search) with the goal of having his car repaired. Note that not all the cells in this line are filled, meaning that, for instance, the instrument used for the discovery activity is not specified. Should we describe a service based (exclusively) on a certain mediator for the discovery process, the name of such mediator would be specified in the "Instrument" cell.

After the service is discovered, the negotiation between customer and mechanic starts; the goal is (probably) an agreement and the negotiation is on the service customization (in other words, how the service type in the job description is tailored to the customer's needs). At that point the mechanic activates the service, i.e., the related scheduling and organization activities. The last two events usually take place in the garage and the whole service acquisition event is performed after the commitment has been taken and before the occurrence of the actual repair.

The actual service process (as can be noted from Figure 1) is a very complex one, consisting of a lot of interconnected activities; here, for simplicity reasons, we choose to represent only the service's core actions.

In the service process event, the mechanic, with his tools and in his garage, performs some actions on the car aimed at having it repaired; this in the interest of the customer.

Finally, there is an articulated service value exchange event, which is constituted by a bunch of activities corresponding to what counts as a "sacrifice" or an "exploitation" from the producer's and customer's points of view. This is a complex topic, that deserves a more

**Table 1: The Garage example**

| | | **Agent** | **Theme/Patient** | **Goal** | **Recipient/ Beneficiary** | **Instrument** | **Location** | **Time/duration....** |
|---|---|---|---|---|---|---|---|---|
| **Service Commitment** | | Garage's owner | Job description | | PA (Chamber of Commerce) | Subscription act | Province/ Region? | Starting at fixed date before, opening, until the duration of the license |
| **Service Acquisition** | **Discovery** | Customer | Mechanic | Car repaired | | | | After opening and before actual repair |
| | **Negotiation** | Customer, Mechanic | Service customization | (Agreement) | | | Garage | |
| | **Activation** | Mechanic | Internal execution plan | | | | Garage | |
| **Service Process** | | Mechanic | Car | Car repaired | Customer | Mechanic's tools | Garage | Period in which the repair actually occurs |
| **Service Value Exchange** | **Producer's sacrifice** | Mechanic | Working hours | Being payed | | | Garage, bank… | A certain time (usually) after that the car has been repaired |
| | **Customer's sacrifice** | Customer | Money, car's unavailability, time needed to pick-up car... | Car repaired | | | | |
| | **Producer's exploitation** | Mechanic | Money | | | | | |
| | **Customer's exploitation** | Customer | Car repaired/car availability | | | | | |

thorough examination, because both the components of cost and those of revenue can be many and different evaluations can be conducted with different purposes. Simplifying a lot, here we can say that the mechanic counts as a sacrifice his working hours with the goal of being paid, while the customer counts as a sacrifice the money he pays, the time to go to the garage, the time the car is unavailable and so on with the goal of having the car repaired; the mechanic earns money, while the customer's revenue consists in having his car available again.

There are some remarks that can be made; first of all, from the knowledge representation point of view, one thing that can be easily observed is that some values must be the same across multiple cells; for instance, the mechanic plays a role of agent in service process, while he plays the role of patient in service acquisition. This might be a problem, as most languages ordinarily used to talk about services (like those based on description logics) are not expressive enough to account for co-reference between variables.

Another remark – a methodological one – is that these tables can be further refined, for example by decomposing the service process event in its internal layers.

Even though the example is quite elementary, it is already possible to see how much additional information the table can convey. The table can also help visualizing the responsibility relations specific to definite events that can be deduced by looking at how the thematic roles in the event are filled. For instance, it is easy to infer from the table that the garage's owner is responsible of the availability of the car repair service, as it is described in the job description according to what it is written in the subscription act signed with the Chamber of Commerce. Similarly, the mechanic is responsible of executing the repair as it has been agreed with the customer in the negotiation phase. As we already noted, the chain of delegation and the transfer of responsibilities are issues that are particularly relevant in the domain of services and the table (with the due refinements) could be a useful tool to visualize all this in a clearer way.

## 5 Concluding Remarks and Future Issues

In this paper we have proposed a novel framework aimed at constituting a common ontological foundation for services science. Let us briefly discuss what the main contributions of this approach are, and what future research directions we are considering.

1. *Revisitation of the difference between internal and external service views.* We have seen that the black box model of services based on external behaviour is too limited, and that a higher expressivity is necessary both to describe services in terms of their internal structure and to properly characterize SLAs and non-functional attributes.

2. *Improvement of the classic definition of services coming from economics.* We have seen that Hill's definition based on change is not general enough, since, for instance, it does not allow to consider services which do not necessarily produce a change, such as fire control.

3. *Focus on core actions instead of pre- and post conditions*. We have seen how pre- and post- conditions cannot by themselves capture important aspects of services, related to the way the service process is performed.

4. *Activity-based service representation.* We have seen how to describe a service in terms of a layered structure of related activities (events, in the most general sense

of this term). The separation of the various activities described in Fig. 1 allows us to properly account for non-functional properties, which instead of generically belonging to the service as a whole are attributes that characterize specific activities. In this way, it is possible to determine what aspect of a given service implementation is responsible for a certain service property. In particular, spatio-temporal attributes can be easily taken into account.

5. *Comprehensive business-oriented approach.* We have introduced a clear distinction between service commitment, service process, and service content, taking also into account important issues affecting service quality and evaluation, such as bundling and presentation activities, acquisition activities, and actions related to the service value chain.

6. *Conceptual analysis of the patterns of responsibilities across services.* We have conducted an analysis that takes into account the legal perspective, which is central with respect to service level agreements.

7. *Common framework to describe service according to different views,* in terms of more or less general constraints among the various service activities, providing an ontological foundation to the technique of *responsibility tables* introduced by Alter.

Given the preliminary nature of the present paper, many are the directions in which the analysis can be extended and enriched.

For sure, in order to be effective, this exploratory work needs to result in a formal model, that will constitute an ontology of services that, as a component of a modular social ontology, should be in the end connected with an ontology of organizations.

## Acknowledgements

This work has been carried within the project ICT4Law (ICT Converging on Law: Next Generation Services for Citizens, Enterprises, Public Administration and Policymakers), funded by the Piedmont Region, as well as the project TOCAI.IT (Tecnologie Orientate alla Conoscenza per Aggregazioni di Imprese su InterneT), funded by the Italian Ministry of Research, and the project CSS (Cartella Socio-Sanitaria), funded by the Autonomous Province of Trento. The initial ideas at the basis of this project have emerged from a fruitful collaboration with “Servizio Politiche Sociali e Abitative” of the Autonomous Province of Trento concerning the revision of a catalog of social services, to be shared among different Public Administrations. The first author is funded by a PostDoc grant from the Autonomous Province of Trento.

## References

1. Chesbrough, H. and J. Spohrer, *A Research Manifesto for Services Science.* Communications of the ACM, 2006. **49**(7): p. 35-40.
2. Alter, S., *Service system fundamentals: Work system, value chain, and life cycle.* IBM Systems Journal, 2008. **47**(1): p. 71-85.
3. Baida, Z., *Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling*. 2006, Vrije Universiteit Amsterdam.

4. Janssen, M. and R. Wagenaar. *From Legacy to Modularity: a Roadmap Towards Modular Architectures Using Web Services Technology*. in *Electronic Government*. 2003: Springer.
5. Papazoglou, M.P. and D. Georgakopoulos, *Service-Oriented Computing.* Communications of the ACM, 2003. **46**(10): p. 25-28.
6. Traverso, P. and M. Pistore. *Automated Composition of Semantic Web Services into Executable Processes*. in *International Semantic Web Conference (ISWC'04)*. 2004. Hiroshima, Japan.
7. Vetere, G. and M. Lenzerini, *Models for semantic interoperaility in service-oriented architectures.* IBM Systems Journal, 2005. **44**(4): p. 887-903.
8. Petrie, C. and C. Bussler, *The Myth of Open Web Services: The Rise of the Service Parks.* IEEE Internet Computing, 2008. **12**(3): p. 94-96.
9. Sycara, K., *Unthethering Semantic Web Services*, in *Semantic Web Services, Part 2*, D. Martin and J. Domingue, Editors. 2007, IEEE Intelligent Systems. p. 11-13.
10. Fensel, D. and C. Bussler, *The Web Service Modeling Framework WSMF.* Electronic Commerce Research and Applications, 2002. **1**: p. 113-137.
11. Roman, D., et al., *Web Service Modeling Ontology.* Applied Ontology, 2005. **1**(1): p. 77-106.
12. Weigand, H., et al., *Value-Based Service Modeling and Design: Toward a Unified View of Services* in *Advanced Information Systems Engineering*, P. van Eck, J. Gordijn, and R. Wieringa, Editors. 2009, Springer: Berlin / Heidelberg. p. 410-424.
13. Terlouw, L. and A. Albani, *An Enterprise Ontology-Based Approach to Service Specification.* IEEE Transactions on Services Computing, to appear.
14. O'Sullivan, J., *Towards a Precise Understanding of Service Properties*, in *Faculty of Information Technology*. 2006, Queensland University of Technology. p. 232.
15. Baida, Z., J. Gordijn, and H. Akkermans, *Service Ontology*. 2001, Free University Amsterdam.
16. Dumas, M., et al. *Towards a semantic framework for service description*. in *Data Semantics 9: Semantic Issues in E-Commerce 239*. 2003. Hong Kong: Kluwer.
17. Masolo, C., et al., *The WonderWeb Library of Fundational Ontologies and the DOLCE ontology. WonderWeb Deliverable D18, Final Report (vr. 1.0. 31-12-2003)*. 2003.
18. Castelfranchi, C., *Grounding We-Intention in Individual Social Attitudes: On Social Commitment Again*, in *Realism in Action - Essays in the Philosophy of Social Sciences*, M. Sintonen and K. Miller, Editors. 2003: Dordrecht.
19. Jennings, N.R., *Commitment and conventions: The foundation of coordination in multi-agent systems.* The Knowledge Engineering Review, 1993. **8**(3): p. 223-250.
20. Verdicchio, M. and M. Colombetti. *A logical model of social commitment for agent communication*. in *AAMAS 2003*. 2003: Elsevier.
21. Singh, M.P., *An Ontology for Commitments in Multiagent Systems: Toward a Unification of Normative Concepts.* Artificial Intelligence and Law, 1997. **7**: p. 97-113.
22. Cauvet, C. and G. Guzelian. *Business Process Modeling: A Service-Oriented Approach*. in *HICSS '08, 41st Annual Hawaii International Conference on System Sciences*. 2008: IEEE Computer Society.
23. Hill, T.P., *On Goods and Services.* Review of Income and Wealth, 1977. **23**(4): p. 315–338.
24. McCarty, L.T., *Ownership: A case study in the representation of legal concepts.* Artificial Intellingence and Law, 2002. **10**(1-3): p. 135-161.

25. Parasuraman, A., V.A. Zeithaml, and L.L. Berry, *A Conceptual Model of Service Quality and Its Implications for Future Rersearch.* Journal of Marketing, 1985. **49**(4): p. 41-50.
26. Sasser, W.E.J., R.P. Olsen, and D.d. Wyckoff, *Management of Service Operations: Text and Cases*. 1978, Boston, MA: Allyn & Bacon.
27. Gronroos, C., *A Service-Oriented Approach to Marketing of Services.* European Journal of Marketing, 1978. **12**(8): p. 588-601.
28. Lehtinen, U. and J.R. Lehtinen, *Service Quality: A Study of Quality Dimensions*. 1982, Service Management Institute: Helsinki.
29. Lucy, W., *Philosophy of Private Law*. 2007, Oxford: Oxford University Press.
30. Hohfeld, W.N., *Some Fundamental Legal Conceptions as Applied in Judicial Reasoning.* The Yale Law Journal, 1913. **23**(1): p. 16-59.
31. Beladiez Rojo, M., *Responsabilidad e imputación de danos por el funcionamiento de los servicios públicos*. 1997, Madrid: Tecnos.
32. Perez Moreno, A., *Responsabilidad en la gestión indirecta de obras y servicios públicos.* , in *La responsabilidad patrimonial de los poderes públicos*, Martínez and Calonge, Editors. 1999, Marcial Pons: Madrid, Barcelona. p. 399-418.
33. Becker, H., *Notes on the Concept of Commitment.* American Journal of Sociology, 1960. **LXVI** p. 32–40.
34. Falcone, R. and C. Castelfranchi, *The human in the loop of a delegated agent: the theory of adjustable social autonomy.* IEEE Transactions on Systems, Man, and Cybernetics, Part A, 2001. **31**(5): p. 406-418.