Foundations for OWL-S: Aligning OWL-S to DOLCE

Peter Mika and **Marta Sabou**
Departments of Business Informatics
and Artificial Intelligence
Vrije Universiteit Amsterdam
The Netherlands
pmika@cs.vu.nl, marta@cs.vu.nl

Aldo Gangemi
Laboratory for Applied Ontology
National Research Council Rome
Italy
gangemi@ip.rm.cnr.it

Daniel Oberle
Institute AIFB
University of Karlsruhe
Germany
oberle@aifb.uni-karlsruhe.de

Abstract

Clarity in semantics and a rich formalization of this semantics are important requirements for ontologies designed to be deployed in large-scale, open, distributed systems such as the envisioned Semantic Web. This is especially true for the description of web services, which should enable complex tasks involving multiple agents. As one of the first initiatives of the Semantic Web community for describing web services, OWL-S attracts a lot of interest and increases its user base even though it is still under development. Our contribution to the development of this ontology is to identify some of its problematic aspects and to suggest enhancements through alignment to a foundational ontology. However, the contribution of our work is not limited to the concrete results reported in this paper, but rather consists of examples of the benefits of alignment to foundational ontologies and a description of the method itself.

Introduction

Ontologies are the basic infrastructure for the Semantic Web whose idea hinges on the possibility to use shared vocabularies for describing resource content and capabilities. Clarity in semantics and a rich formalization of this semantics are important requirements for ontologies designed to be deployed in large-scale, open, distributed systems such as the envisioned Semantic Web. This is due to the fact that ontologies are used to negotiate meaning, either for enabling effective cooperation between multiple artificial agents, or for establishing consensus in a mixed society where artificial agents cooperate with human beings. Foundational ontologies are typically used to fulfill those requirements. They serve as a starting point for building new domain and application ontologies, provide a reference point for easy and rigorous comparisons among different ontological approaches and create a framework for analyzing, harmonizing and integrating existing ontologies and metadata.

All of the above especially holds for ontologies for the description of web services because they enable complex tasks involving multiple agents. Web services are becoming ever more important resources on the Web and standards are being developed for low-level descriptions of web services (Christensen *et al.* 2003). As one of the first initiatives of the Semantic Web community for semantically describing web services, OWL-S (Martin *et al.* 2003), formerly known as DAML-S, attracts a lot of interest and increases its user base even if still under development. OWL-S is an ontology of general concepts allowing automatic discovery, composition and invocation of web services.

Our contribution to the development of this ontology is to identify some of its problematic aspects and to suggest enhancements through alignment to a foundational ontology. We found that OWL-S suffers conceptual ambiguity, lacks concise axiomatization, is designed too loosely and does not allow to distinguish objects within an information system and the real world. With our alignment, we have obtained significant results for enhancing these aspects of the ontology. We present these results for the benefit of the designers and users of OWL-S. Furthermore, we developed a Core Ontology of Services as a middle layer which can also be used for aligning other (web) service description languages. Lastly, we note that the contribution of our work is not limited to the concrete results reported in this paper, but rather consists of (1) examples of the benefits of alignment to foundational ontologies and (2) a description of the method itself.

The paper is structured as follows. We start with a motivating scenario followed by related work. We identify and explain several problematic aspects of OWL-S which can also be seen as a motivation for our work. The subsequent Section presents the main body of work, viz. the alignment of OWL-S to the DOLCE foundational ontology. It also includes a short introduction to the foundational ontology itself. After that we revisit the problematic aspects introduced before and detail how they can be solved by the alignment and conclude.

Motivating example

To motivate our case, we present a fictitious, but realistic setting for a service. Below we describe this service from two different viewpoints.

Context 1 Marta would like to surprise her Italian boyfriend with a nice pasta dinner for their anniversary next week. For this special occasion, Marta considers to find new pasta recipes by looking for information online or buying a cookbook, either in a bookshop or online. Besides timing, Marta has some constraints on her choice such as the amount of money she is willing to spend.

Context 2 Amazon.com is a web service that provides its users with an interface where books can be located based on their descriptions, ordered and paid. The books are delivered to the user through the postal service.

Firstly, one can note that both descriptions naturally cross the boundary between an information system (with objects such as a record about a book) and the external world (with objects such as the physical book). The reason is that the web service of Amazon.com is only a part of the overall service to which a value is attributed by the requestor. We believe that this phenomenon will characterize most real world services, where users are paying not simply for their information being recorded and manipulated, but for the overall process, which includes actual changes and effects in the real world, such as a book being delivered on time to Marta.

Secondly, both of these descriptions are independent views or descriptions of a world (real or imagined) and indeed significantly differ in the notions that are used (e.g. recipe or information vs. book) and the granularity of the descriptions (high level tasks vs. detailed processes). Similarities are to be found on the level of *constructs* used to describe these views: both of them discuss roles that can be played by a number of objects (e.g. information about pasta recipes can be in the form of books or online texts) and plans or courses of events which can be realized by different sequences of activities.

In this paper we discuss how these constructs are captured in an abstract form in the Descriptions & Situations ontology. We will also show how we can use this ontology as the backbone for our Core Ontology of Services, which is then used to align OWL-S to the DOLCE foundational ontology. Besides a disambiguation of OWL-S, the outcome of this process are suggestions for potential improvements on the current design of the ontology.

Related Work

Previous efforts responded to some of the problems of OWL-S. We briefly discuss the two initiatives we are aware of by describing their motivation, the parts of OWL-S they focus on, the techniques they use as well as some initial results (when available).

The first initiative (Narayanan & McIlraith 2003) is motivated by the need of formal semantics to describe, simulate, automatically compose, test and verify web service compositions. It focuses solely on the OWL-S Process Model which provides all the constructs for specifying composition. The authors establish a situation calculus semantics for the main elements in the OWL-S *ServiceModel* (e.g. atomic and composite processes, conditional effects and outputs), then translate it to the operational semantics provided by Petri Nets. Indeed, this semantics allowed to re-use an existing simulation and modelling environment. Further, the authors were able to identify more tractable subsets of OWL-S (less expressive but more efficient analysis for verification, composition and model checking).

The second effort (Ankolekar, Huch, & Sycara 2002) also focuses only on the OWL-S *ServiceModel* and proposes a concurrent operational semantics that incorporates subtype polymorphism. The motivation for this work is to provide an initial reference semantics that would discover any possible ambiguity in the developed language. It would also serve for developing techniques for automated verification of OWL-S models. Finally, if other web standards would provide a similar semantics it would be much easier to compare them and to understand their strengths and weaknesses. The authors of both efforts mutually acknowledge the similarity between the two proposed semantics, except some minor details discussed in (Ankolekar, Huch, & Sycara 2002).

Besides aiming at increased formal axiomatization, we wish to explain the OWL-S concepts in terms of a foundational ontology which reflects several generally accepted theories from linguistics, psychology, human cognition etc. We show that this "ontological" analysis of OWL-S also brings to surface several irregularities in the model (just like the reference semantics promises to do). Further, one of the long term benefits of alignment is that it allows a comparison between several aligned ontologies (a goal also stated in (Ankolekar, Huch, & Sycara 2002)). As a result we extend our analysis to the entire OWL-S model. From a methodological perspective, the previous approaches provide independent reconstructions of OWL-S, while, through alignment, we embed the OWL-S model in the larger context offered by the foundational ontology. Therefore we can deduce that OWL-S does not address the difference between a real life object (e.g. book) and its representational counterpart in an information system (e.g. ISBN number), an important ontological distinction. Finally, the semantics established by previous work are not reflected in the current OWL formalization of the model. In our case, the model inherits all the axiomatization available for DOLCE.

Problematic aspects of OWL-S

This Section identifies and illustrates some of the problematic aspects of understanding OWL-S and the kinds of difficulties that one would encounter representing the motivating example shown before. Our goal here is to raise awareness to these issues, most of which we revisit later when discussing some of our solutions. We also relate these issues to the question of ontology quality.

Ontology quality is the topic of (Borgo *et al.* 2002), which provides (among others) three criteria for evaluation: *extensional coverage* (concerning the amount of entities that are supposed to be described by an ontological theory), *intensional coverage* (concerning what kinds of entities are described by an ontological theory), and *precision* (concerning what axioms are required to describe just the models the ontology designer intends to cover). According to these criteria, a good ontology should approximate the domain of discourse that is supposed to be described, it should have a signature that maps all the kinds of entities intended by the designer, and it should axiomatize the predicates in order to: 1) catch all the intended models, and 2) exclude the unintended ones.

Below we introduce four problems encountered in OWL-S. The first one (conceptual ambiguity) features both insufficient intensional coverage and overprecision. The second and the third (poor axiomatization and loose design) are cases of insufficient precision. In the third problem, the weakness is mainly inherited by limitations of OWL expressivity. The fourth (narrow scope) is a case of both extensional and intensional coverage.

Conceptual Ambiguity

Since there is no clear conceptual framework behind OWL-S, it is often difficult for users to understand the intended meaning of some concepts, the relationship between these concepts as well as how they relate to the modelled services. Many concepts are still being clarified both within the OWL-S committee and in public mailing lists[1]. The Web Services Architecture Working Group[2] of the W3C is also expected to put forward a conceptual architecture of web services. Unfortunately, this framework at present provides only a natural text description of concepts and therefore no formal links can be established to the similar concepts used in OWL-S.

Conceptual ambiguity affects particularly the upper level of OWL-S. The notion of a service is introduced in (Martin *et al.* 2003) as follows: "By 'service' we mean Web sites that do not merely provide static information but allow one to effect some action or change in the world, such as the sale of a product or the control of a physical device". Later, we read that "any Web-accessible program/sensor/device that is declared as a service will be regarded as a service".

However, neither of these definitions are operationalized as neither the concept of a "website" nor the "Web" appears in the ontology. Instead, the notion of a service is characterized solely by its relationship to a number of *ServiceProfiles*, at most one *ServiceModel* and any number of *ServiceGroundings*, which is not sufficient to understand the concept of *Service* considered by OWL-S.

We note that the term web service and closely related terms (e-service, service, etc.) also suffer from overloading. In our search for possible formalizations, we found a variety of definitions emphasizing different aspects of a service (Gangemi *et al.* 2003): offering functionality (usefulness for a particular task), interoperability using standards or providing an interface to an existing system. We also refer the reader to the work of Baida et al. (Baida, Gordijn, & Akkermans 2004), which compares and contrasts the definitions used in the business literature, in software engineering and in information sciences.

Poor axiomatization

As OWL-S is aimed to be machine processable and operates in an open environment, it is important that each concept is characterized by a rich axiomatization in order to support meaningful inferences. In general, we believe that the level of commitment in OWL-S will need to be raised if it is to support the complex tasks put forward by the coalition.

Unlike the issue mentioned in the previous section, poor axiomatization reflects the lesser problem when the definition of concepts is conceptually clear, but axiomatization in the ontology itself needs improvement. In many respects, OWL-S shows the characteristics of a typical application ontology: there is no firm concept or relation hierarchy (most concepts and relations are direct subconcepts of the top level concept or relation) and several relations take *owl:Thing* as their domain or range.

We propose that by adding foundations to OWL-S, the level of axiomatization can be raised significantly. Alignment to a foundational ontology means relating the concepts and relations of an ontology to the basic categories of philosophy, linguistics or human cognition. This approach has the advantage that restrictions on the level of common sense are inherited by the concepts in the application ontology. This prompts the ontology engineer to sharpen his notions with respect to the distinctions made in the foundational ontology. It also promotes reuse by highlighting commonalities, which especially helps to reduce the proliferation of relations that is typical for application ontologies.

Alignment to a well-modularized foundational ontology also allows to selectively import theories from the ontology such as mereology, time theory etc. We will demonstrate this later during alignment of the control constructs of OWL-S to the Ontology of Plans which is part of the DOLCE foundational ontology.

Loose design

A further problematic aspect of OWL-S from an ontologist's point of view is an entangled design. At the heart of this problem lies the purpose of OWL-S in providing descriptions of various views on web services required to support a number of different service related tasks (discovery, composition, invocation). Besides the functional dimension, our motivating example shows the clear need for an ability to contextualize our descriptions to represent various points of view on a service, possibly with different granularity.[3] Most of these views, however, are overlapping in that they concern some of the same attributes of a service.

A straightforward modularization in such cases results in an entangled ontology, where the placement of certain knowledge becomes arbitrary and intensive mapping is required between modules. This phenomenon is well known in object-orientation, where the notion of *aspects* (Elrad, Filman, & Bader 2001) was proposed to encapsulate concerns that cross-cut the concept hierarchy of software.

A case in point is the application of attribute binding in OWL-S which is necessary for expressing that the output of one process is the input for another process or that the output of a composite process is the same as the output of one of its subprocesses. In programming, such equivalences are expressed by the use of *variables*. Variables are governed by the rules of *scoping*, which define the boundaries of commitment.

[1] cf. `http://www.daml.org/services`

[2] cf. `http://www.w3.org/2002/ws/arch`

[3] The OWL-S specification mentions the ability to use *Profiles* for providing such views. However, no actual constructs are provided to map them to possible service executions or to each other.

Lacking variables, argument binding is expressed in OWL-S by explicit value maps. As shown in Figure 1, the value map has the form of a *List*, attached to a *ProcessComponent*. This *List* should contain instances of the *ValueOf* concept as members[4]. Each *ValueOf* concept should point to a single relation of a single concept by using *theProperty* and *atClass* relations[5]. For example, in case of two processes A and B where process B takes the output of process A as an input, the list would have two *ValueOf* members, one related to concept A and the output relation, while the other would be related to concept B and its input relation.

The reader may also note that the intended meaning of the entire construct, namely that all 'sensible' instantiations of the process should respect the equivalences expressed in the value map, is not encoded in the axiomatization. This is due to the lack of expressivity of the Description Logic used.

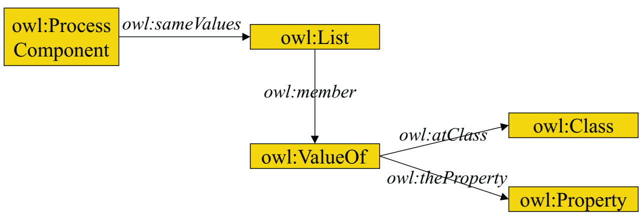

Figure 1: The representation of attribute binding in OWL-S

Besides a tedious representation, an unfortunate consequence of the present solution is that we can only guess about the scope of the commitment represented by the value map. OWL-S seems to suggest attaching the value map to the process whose sub-processes are involved in the value map. As argued above, however, there could be multiple value map restrictions on the inputs/outputs of a process resulting from service composition (expanding/collapsing processes). Taking the current OWL-S proposal, it is unclear how one could approach such situation.

Narrow scope

As shown by our motivating example, the scope of OWL-S needs to be extended to represent real world services that naturally cross the lines between information systems and the physical world. While OWL-S acknowledges this aspect of services, it is unclear how a distinction could be made in OWL-S between the objects and events within an information system (regarding data and the manipulation of data) and the real world objects and events external to such a system. Using a foundational ontology, however, it is possible and even required for the creator of a description to make such distinctions, because they fundamentally affect the ontological nature of the objects and events concerned.

Besides its insufficient intensional coverage, the OWL-S core also shows an overcommitment in precision: the top *Service* concept is related to the *ServiceModel* concept with a cardinality 1:1. This means that for each *Service*, only one *ServiceModel* is expected to hold. This prevents us to consider alternative *ServiceModels*, or to evaluate the relationship between a *ServiceModel* required by a customer's guideline, or by a legal regulation, and the one underlying the provider's system, for instance.

A further contribution of our work is to extend OWL-S with relationships for mapping between service descriptions and the elements of actual service executions, which are not yet covered by OWL-S. These relationships will be directly inherited from the Descriptions & Situations ontology.

Alignment

This Section shows how we align OWL-S to the DOLCE foundational ontology which is extended by the Descriptions & Situations plug-in. In addition, we come up with a Core Ontology of Services and briefly depict how OWL-S' concepts are to be expressed by using it. Finally, we give a short summary of the methodology.

DOLCE

Foundational ontologies are conceptualizations that contain specifications of domain independent concepts and relations based on formal principles derived from linguistics, philosophy, and mathematics. DOLCE, a Descriptive Ontology for Linguistic and Cognitive Engineering, is the first module of a future foundational ontology library and designed to be minimal in that it includes only the most reusable and widely applicable upper-level categories, rigorous in terms of axiomatization and extensively researched and documented (Oltramari *et al.* 2002; Masolo *et al.* 2002).

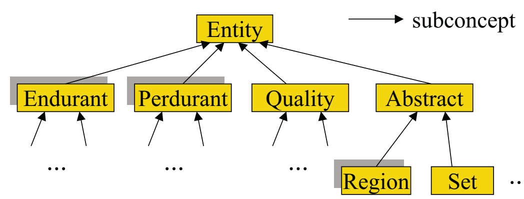

Figure 2: The top-level taxonomy of DOLCE.

The upper part of DOLCE's taxonomy is sketched in Figure 2. DOLCE is based on a fundamental distinction between enduring and perduring entities. The main relation between *Endurants* and *Perdurants* is that of participation: an endurant "lives" in time by participating in a perdurant. For example, a person, which is an endurant, may participate in a discussion, which is a perdurant. A person's life is also a perdurant, in which a person participates throughout all its duration. *Qualities* can be seen as the basic entities we can perceive or measure: shapes, colors, sizes, sounds, etc. Finally, *Abstracts* do not have spatial or temporal qualities, and they are not qualities themselves.

Although DOLCE is axiomatized in first order logic and specified in the LOOM language (MacGregor 1991), less expressive languages, in particular OWL DL (McGuinness & van Harmelen 2003), are possible. The basic strategy is to

[4]However, this is not enforced. There's also no explanation about what the ordering means.

[5]The cardinality restrictions are missing from the formalization.

isolate the part of the axiomatization that can be expressed in OWL.[6]

Descriptions & Situations

While modelling physical objects or events in DOLCE is quite straightforward, intuition comes to odds when we want to model non-physical objects such as plans, roles or parameters. This difficulty is due to the fact that the intended meaning of non physical objects results from statements, i.e. their meaning emerges only in the combination of other entities. On the other hand, non physical objects may change and be manipulated similar to physical entities, and are often treated as first-order objects. That means an ontology should account for such objects by modelling the context on which they depend. The representation of context is a common problem in many realistic domains from technology and society (such as law or finance) which are full of non physical objects.

In order to respond to those modelling requirements, we developed a plug-in to DOLCE, called Descriptions & Situations (D & S) (Gangemi & Mika 2003). D & S results to be a theory of ontological contexts because it is capable of describing various notions of context (physical and non physical situations, topics, plans, beliefs, etc.) as entities. Like DOLCE, it features an extensive and philosophically concise axiomatization.

First of all D & S defines descriptive components such as *Parameters*, *Functional Roles* and *Courses of Events*. Rules enforce that each descriptive component links to a certain category of DOLCE like depicted in Figure 3. *Parameters* are *valued-by Region* or *Functional Roles* are *played-by Endurants*, for instance.

Second, the entities along their links (provided by the descriptive components) constitute a certain situation. Accordingly, D & S distinguishes between the elements of a description and the entities that are described. The latter are the elements of a situation (regions, endurants and perdurants) which are constrained by the parameters, roles and courses of a description. In Figure 3 the respective concepts are thus grouped into a *Situation* and a so-called *Situation Description* (*S-Description*, or *Context*).

D & S shows its practical value when applied as a *design pattern* for (re)structuring application ontologies that require contextualization. As we will see in the remainder of this section, this is the case when describing (web) services.

A Core Ontology of Services

The descriptions of services show a clear contextual nature and are to be modelled as *Situation Descriptions* in the sense of DOLCE and Descriptions & Situations. One may only have to consider the number of different views that may exist on a service: the view of a service provider, that of the service requestor or the legal view of a contract etc. The concepts used to formulate any given view are clearly separate from the actual objects they act upon and often independent from the concepts appearing in other views.

[6]The OWL version of DOLCE can be found at `http://www.isib.cnr.it/infor/ontology/DOLCE.html`

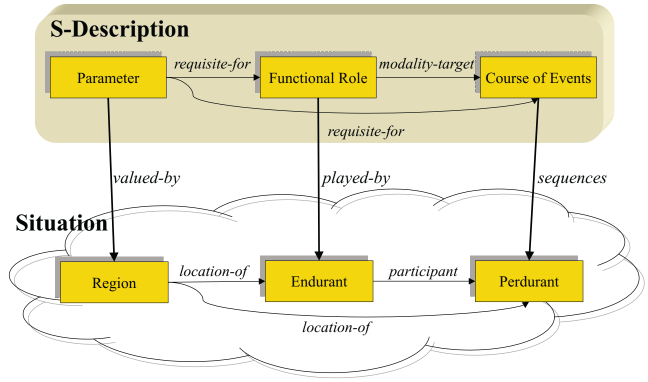

Figure 3: Descriptions and Situations

Different views on the service need not be equally detailed either. For example commercial advertisements typically feature only selected characteristics of a service. The various views also naturally focus on different aspects of a service, which means that the descriptions may only be partially mapped to each other.

Instead of directly aligning OWL-S to Descriptions & Situations, we developed a Core Ontology of Services (COS) and aligned the sources to this ontology. This two-stage alignment is a common technique when the conceptual gap between the source ontologies and the foundational ontology is large. The Core Ontology of Services also features a concise axiomatization detailed in (Gangemi *et al.* 2003) and can be reused in other scenarios (e.g. purely commercial services).

Currently, we have considered five frequently occurring descriptions of a service, where each is a separate viewpoint of the same service: (Service) Offering, Request, Agreement, Assessment and Norms (more views may be added in the future when needs arise). In this work we will only detail the Service Offering view.

All service views are specializations of *Situation Description* defined in the Descriptions & Situations ontology. In addition, we divide DOLCE's *Task* into *Computational Task* and *Service Task*. This allows us to model activities in an information system and in the real world. Our Core Ontology of Services may optionally take advantage of a number of concepts from the Ontology of Plans which is another plug-in to DOLCE. It allows the division of tasks into elementary and complex and the construction of complex tasks from elementary ones among other features. DOLCE's *Agentive Functional Role* and *Instrumentality Role* are subdivided in *Requestor*, *Provider*, *Executor* and in *Input*, *Output*, respectively. *Computational Input* and *Computational Output* are kinds of input and output that are played only by information objects and only have exploitation within *Computational Tasks*. Note that there are other features in this ontology which are neglected here due to the lack of space.

Figure 4 shows a simplified example of how the Core Ontology of Services can be used to create different service views that act on the same setting. In this enactment of our motivating example, both the *Delivery Item* concept of the

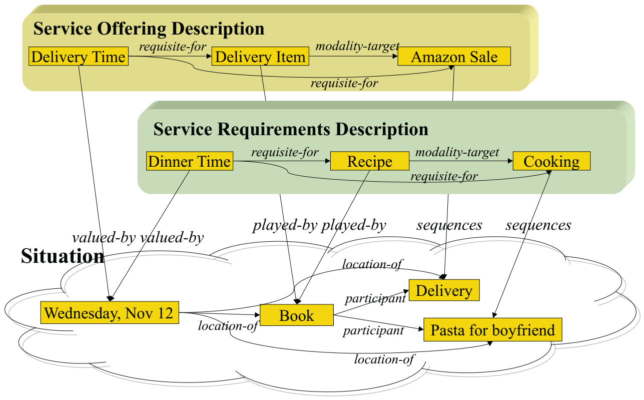

Figure 4: Multiple descriptions of a service using the Core Ontology of Services.

service provider and the *Recipe* concept of the service requestor are *played by* the same *Book*, which participates in both the *Delivery* and the *Cooking* activities. The order has a parameter of a *Delivery Time*, which is also mapped to the dinner time parameter of the *Cooking* activity: both have to be *valued-by* the same or overlapping time regions for the service to be useful to Marta and her boyfriend.

Aligning OWL-S to the Core Ontology of Services

In the following we briefly describe the alignment of OWL-S to the Core Ontology of Services.

Although the definition of a service is ambiguous even in the natural text description of OWL-S, for the sake of argument we considered an *owl-s:Service* as a Service Offering Description, which has the *ServiceProfile* and *ServiceModel* (also Service Offering Descriptions) as parts. Actors in the *ServiceProfile* are aligned as Agentive Functional Roles . The *ServiceModel* concept was aligned to our Service Task concept, while the individual control constructs were mapped to task components provided by the Ontology of Plans.

In the Core Ontology of Services, the notions of Inputs and Outputs were modelled as Non-Agentive Functional Roles and not as relations in OWL-S. Nevertheless, alignment was possible by means of a composed relationship. On the other hand, the notion of preconditions and effects are inherited from the Ontology of Plans (task-precondition and task-postcondition) where they are modelled as Situations.

As it was not related to the focus of work, we omitted the alignment of the particular grounding ontology for WSDL (Christensen *et al.* 2003). Nevertheless, the notion of *Software* is present in the Core Ontology of Services as *Information Object* that can be expressed according to any number description systems. WSDL could be considered as such a description system and modelled to the extent required to express groundings. [7]

[7]The Core Ontology of Services and the OWL-S alignment are available for download at `http://www.cs.vu.nl/~pmika/research/www2003/`

Summary

The ontology stack in Figure 5 summarizes our alignment effort. We used DOLCE as foundational ontology, extended it by the Descriptions & Situations plug-in and defined our Core Ontology of Services, which was used to align OWL-S. Note that this methodology of alignment has been used to align and compare other service description efforts as well, e.g. the Web Services Architecture (WSA) or the ontology used within the Application Server for the Semantic Web (both alignments are detailed in (Gangemi *et al.* 2003)). Specialized domain and application ontologies of service descriptions are formulated according to one of these generic service ontologies.

Our method was a combination of a bottom-up and a top-down approach. On the one hand, ontologies in the lower layers provided representation requirements for the higher layers, which abstracted their concepts and relationships. On the other hand, the upper layers provided design guidelines to the lower layers. This also meant that although our goal was to preserve the structure of OWL-S as much as possible, our method suggested a rearrangement of the ontology based on the backbone provided by the Descriptions & Situations ontology.

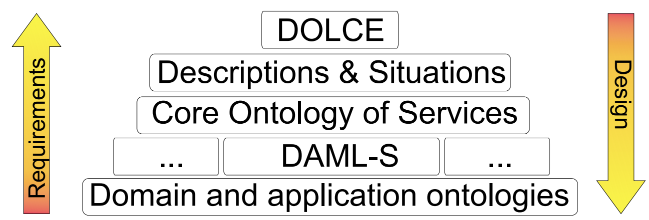

Figure 5: Ontology stack

Solutions

This Section highlights that the aforementioned problematic aspects of OWL-S can be solved by our alignment.

Conceptual Disambiguations

The alignment to a foundational ontology helped us in understanding and crystalizing several concepts of OWL-S. As an example, ontological analysis explained the difference between an information object, its application domain counterpart and the role it plays in an information system. This indicated possible enhanced modelling: since the same information object is modelled both in the *ServiceProfile* and *ServiceModel*, it is more logical to consider a single instance playing multiple roles. This improvement is already considered by the OWL-S Coalition.

In our Core Ontology of Services, we went further to separate the functionality, process and software aspects of a service loaded onto the single concept of *Service* in OWL-S. It replaces *Service* with the concept of different kinds of Service Descriptions, which are S-Descriptions (a context in D & S) that envision a process as well as certain roles related to the individual tasks of the process. Inputs, outputs and abstract tools used to carry out a certain task are examples of roles. In case of information services, inputs and outputs are played by information objects and tools are played by particular software implementations.

While this definition of a Service Description may not be the only one, the fact that it is formulated according to a foundational ontology allows to compare it to alternative definitions and foster discussion on alternative conceptualizations of a (semantic) web service.

Increased Axiomatization

A key advantage of the alignment to a foundational ontology is that it prompts the engineer to take a stance with respect to the principles established by the foundational ontology. What is typically gained is an increased understanding of one's own ontology and a richer axiomatization through ties to the foundational ontology. DOLCE mitigates the danger of overcommitment in this process (importing theories that are not used or not shared by the engineer) by extensive modularization along world views (3D, 4D, etc.) and domains (law, finance, etc.).

As an example, in our Core Ontology of Services we have made use of an Ontology of Plans which includes subtypes of the generic Task concept for a detailed modelling of plans or process models. These constructs are directly comparable to the control constructs of OWL-S, but provide a higher level of axiomatization. An example of such types is DOLCE's *Synchro-Task* that matches the concept of "join" in the "Split-Join" control construct from OWL-S. A synchronization task is typically used to bind the execution of a "planning" activity rather than of a domain activity, since the referred activity is supposed to re-synchronize a process when it waits for the execution of two or more concurrent (or partly concurrent) activities.

Higher axiomatization is partly possible by the natural linkage to the Ontology of Time, another plug-in to DOLCE, for describing (constraints on) temporal relations between process elements when they are executions of a plan. OWL-S would also need such an Ontology of Time and then it would be natural to adopt or reference an existing ontology instead of creating an ontology from scratch.

The Ontology of Plans also allowed to align relations such as *owl-s:components*, which is used to relate control constructs to their components. In OWL-S this relation is described merely as a subrelation of *owl:Property* with a domain of *ControlConstruct*. In our work, we aligned this relation to the *temporary-component* relation in DOLCE. The latter has a firm foundation as a subrelation of the more basic *partly-compresent-with* and *component* relations and is also characterized in terms of restrictions on its application to other basic concepts, such as *Object*, *Description*, *Event*, etc.

Improved design

In our work we propose to complement modularization in OWL-S with contextualization as a design pattern. Contextualization allows us to move from a monolithic process description of a service to the representation of different, possibly conflicting views with various granularity. The Descriptions & Situations ontology provides us the basic primitives of context modelling such as the notion of roles, which allows us to talk of inputs and outputs on the abstract level, i.e. independent of the objects that play such roles.

Using this pattern results in a much more intuitive representation of attribute binding, with clearly defined semantics and scoping provided by Descriptions & Situations. Inputs and outputs can be modelled as Functional Roles (more precisely: Instrumentality Roles), which serve as variables in our ontology. A single endurant — for example, a physical book — can play multiple roles within the same or different descriptions and thus it is natural to express that the given book is output with respect to one process, but input to another (see also Figure 4). Moreover, it is easier to represent the requirement that the input of a process *has to be* played by the same instance as the output of another process by putting constraints on the *objects* (and not the process or task) which play these roles (however, the expressivity required is the same and therefore goes beyond the power of OWL).

Besides a more intuitive representation, Functional Roles as components have an explicit scope, namely the S-Descriptions they belong to. Although not addressed in the present work, clearly defined limits in scope are necessary to describe semantic relationships among (service) descriptions, e.g. to talk of conflicts between descriptions.

Wider scope

As we have seen before, web services exist on the boundary of the world inside an information system (Infolandia) and the external world. Except for the rare case of a pure information service, web services carry out operations to *support* a real world service. Functionality, which is an essential property of a service, then arises from the entire process that comprises computational as well as real wold activities.

Web service descriptions are thus necessarily descriptions of two parallel worlds. In Infolandia, the world consist of software manipulating (representations of) information objects. Activities are sequenced by computational processes. Meantime in the real world books are being delivered to their destinations.

The connection between these worlds is that some of the information objects in Infolandia are representations of real world objects. Also, computational activities comprise part of the service execution in the real world. For example, an order needs to be entered by the web agent into an information system, so that the warehouse knows which books to deliver to a given address.

The distinction between information objects, events and physical ones is not explicitly made in OWL-S. Nevertheless, we believe that this distinction is important for disambiguating the nature of services in an open environment such as the Semantic Web. In our work this separation naturally follows from the use of the DOLCE foundational ontology, where the distinction is an important part of the characterization of concepts. In particular, it makes possible to be more precise about the kinds of relationships that can occur among objects or between objects and events.

For example, using DOLCE we can distinguish between a physical object (such as a Person), an information ob-

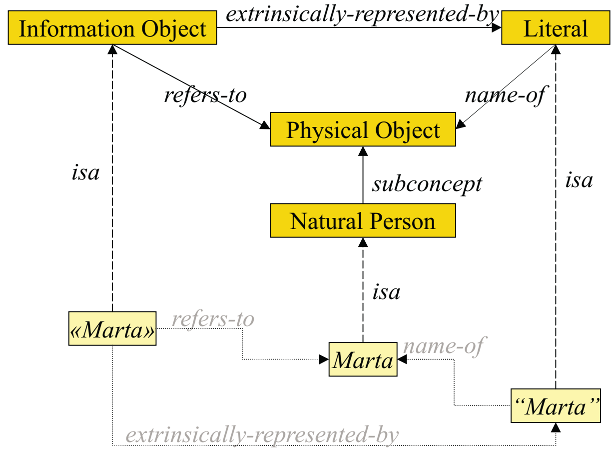

Figure 6: Information Objects in DOLCE

ject (such as the name of a person) and a representation of such information using a particular description system (e.g. a string encoding). The relations provided by DOLCE are shown in Figure 6.

Note that we also relax the intentional definition of OWL-S to cover the case of multiple descriptions of a service in various contexts. Based on this we may define the notion of (a partial or complete) *match* between descriptions as the mapping from the descriptions to a possible model of the world that can be (in part or whole) interpreted as a satisfying situation for both contexts. Future work may include studying other possible relations between service descriptions (inclusion, conflict etc.), potentially based on an investigation of relations between S-Descriptions.

Conclusion

The paper identified several problematic aspects of OWL-S and showed how they can be solved by an alignment to a foundational ontology. We used a stack of ontologies for the alignment made up of DOLCE, Descriptions & Situations as well as the Core Ontology of Services. Note, that the alignment is not dependent on DOLCE, because Descriptions & Situations may be plugged into any foundational ontology. Parts of the service description that deal with service quality and assessment are left as future work.

Our exercise of giving an ontological foundation to OWL-S is useful both for better understanding OWL-S and enriching it with additional formal semantics. We see the presented results as an example of the benefits of alignment to foundational ontologies as our methodology is applicable also to other standards.

We are aware of the fact that the ontology engineer to some extent needs to understand the principles behind the foundational ontology stemming from other sciences: philosophy, psychology, semiotics, communication theory etc. In other words, a (re)engineering of this kind requires a considerable intellectual investment from the knowledge engineer. We think, however, that this investment, materialized in the Core Ontology of Services, will pay off whenever new (web) service ontologies are to be aligned.

Nevertheless, we will try to make the foundational ontology and the alignment process more accessible in the future. We also believe that more efficient tool support for working with large, conceptually rich ontologies will once take much of the burden away from the knowledge engineer.

References

Ankolekar, A.; Huch, F.; and Sycara, K. 2002. Concurrent Execution Semantics for DAML-S with Subtypes. In *1st Int. Semantic Web Conference (ISWC), Proceedings*, volume 2342 of *LNCS*. Springer.

Baida, Z.; Gordijn, J.; and Akkermans, H. 2004. A Shared Terminology for Online Service Provisioning. Available via `http://www.cs.vu.nl/~ziv`.

Borgo, S.; Gangemi, A.; Guarino, N.; Masolo, C.; and Oltramari, A. 2002. Ontology RoadMap. WonderWeb Deliverable D15.

Christensen, E.; Curbera, F.; Meredith, G.; and Weerawarana, S. 2003. Web services description language (WSDL). W3C Note.

Elrad, T.; Filman, R. E.; and Bader, A. 2001. Aspect-oriented programming: Introduction. *Communications of the ACM* 44(10):29–32.

Gangemi, A., and Mika, P. 2003. Understanding the semantic web through descriptions and situations. In *2003 Confederated Int. Conferences DOA, CoopIS and ODBASE, Proceedings*, LNCS. Springer.

Gangemi, A.; Mika, P.; Sabou, M.; and Oberle, D. 2003. An Ontology of Services and Service Descriptions. Technical report, Laboratory for Applied Ontology, National Research Council, I-00137 Rome, Italy. `http://www.isib.cnr.it`.

MacGregor, R. M. 1991. Using a description classifier to enhance deductive inference. In *Proceedings Seventh IEEE Conference on AI Applications*, 141–147.

Martin, D.; Burstein, M.; Denker, G.; Hobbs, J.; Kagal, L.; Lassila, O.; McDermott, D.; McIlraith, S.; Paolucci, M.; Parsia, B.; Payne, T.; Sabou, M.; Sirin, E.; Solanki, M.; Srinivasan, N.; and Sycara, K. 2003. DAML-S (and OWL-S) 0.9 draft release.

Masolo, C.; Borgo, S.; Gangemi, A.; Guarino, N.; Oltramari, A.; and Schneider, L. 2002. The WonderWeb library of foundational ontologies. WonderWeb Deliverable D17. `http://wonderweb.semanticweb.org`.

McGuinness, D. L., and van Harmelen, F. 2003. Web ontology language (OWL) overview. W3C Candidate Recommendation.

Narayanan, S., and McIlraith, S. 2003. Analysis and simulation of Web Services. *Computer Networks* 42(5):675–693.

Oltramari, A.; Gangemi, A.; Guarino, N.; and Masolo, C. 2002. Sweetening ontologies with DOLCE. In *13th Int. Conference, EKAW 2002, Proceedings*, volume 2473 of *LNCS*. Springer.