A Framework for Deriving Semantic Web Services

David Bell[(1)], Sergio de Cesare[(1)], Nicola Iacovelli[(2)], Mark Lycett[(1)], Antonio Merico[(2)]

Corresponding author: Sergio de Cesare (sergio.decesare@brunel.ac.uk)

[(1)] Dept. of Information Systems and Computing
Brunel University
Uxbridge, Middlesex
UB8 3PH, United Kingdom
e-mail: {david.bell, sergio.decesare, mark.lycett}@brunel.ac.uk

[(2)] Svimservice S.p.A.
Via Massaua, 18
70123 Bari
Italy
e-mail: {nicola_iacovelli, antonio_merico}@svimservice.it

A Framework for Deriving Semantic Web Services

Abstract

Web service-based development represents an emerging approach for the development of distributed information systems. Web services have been mainly applied by software practitioners as a means to modularize system functionality that can be offered across a network (e.g., intranet and/or the Internet). Although web services have been predominantly developed as a technical solution for integrating software systems, there is a more business-oriented aspect that developers and enterprises need to deal with in order to benefit from the full potential of web services in an electronic market. This 'ignored' aspect is the representation of the semantics underlying the services themselves as well as the 'things' that the services manage. Currently languages like the Web Services Description Language (WSDL) provide the syntactic means to describe web services, but lack in providing a semantic underpinning. In order to harvest all the benefits of web services technology, a framework has been developed for deriving business semantics from syntactic descriptions of web services. The benefits of such a framework are two-fold. Firstly, the framework provides a way to gradually construct domain ontologies from previously defined technical services. Secondly, the framework enables the migration of syntactically defined web services toward semantic web services. The study follows a design research approach which (1) identifies the problem area and its relevance from an industrial case study and previous research, (2) develops the framework as a design artifact and (3) evaluates the application of the framework through a relevant scenario.

Keywords: Semantic web services, ontological modeling, service content interpretation, scoping, harmonization.

1. Introduction

Web service-based development represents an emerging approach for the development of distributed information systems. Web services are becoming the dominant technique for representing and distributing behavior across multiple systems, even systems that were not initially planned or designed to work together. As a modularization technique, web services mimic the business environment of any market in which economic operators offer and request services as well as provide intermediation.

Although web services have been predominantly developed as a technical solution for integrating software systems, there is a more business-oriented aspect that developers and enterprises need to deal with in order to benefit from the full potential of web services in an electronic market. This 'ignored' aspect is the representation of the semantics underlying the services themselves as well as the 'things' that the services manage. Semantics have business significance because the representation (or description) of services and the 'things' they manage have their real-world counterparts; moreover the outcome of a web service can have real-world effects on the enterprises involved (e.g., creation of an order and its legal implications).

The adoption and diffusion of web services is rapidly growing. This growth is encouraged also by the potential benefits that the emerging Semantic Web can provide. The Semantic Web is intended to be an extension of today's Web (Berners-Lee et al., 2001). While the current Web is fundamentally designed for human use, the Semantic Web aims at achieving a greater degree of communication, coordination and collaboration between computer systems in an autonomous and proactive way for the benefit of people and organizations. In the world of the Semantic Web, ontologies and web services play a key role (Burnstein, 2004).

Ontologies model a domain in terms of the 'things' whose existence can be acknowledged by that domain (Honderich, 1995; Guarino, 1998). Ontologies provide the means for semantically describing web resources, allowing web agents to share a

common knowledge and understanding of available resources and what these resources refer to. Web services must however be discoverable; in other words what a service offers must be described in a public, shared and precise manner. The terms 'public', 'shared' and 'precise' go hand in hand. For a web service (or any web resource) to be publicly identifiable it must be represented in a shared way (i.e., through ontologies) and for shared representations to be accepted by the present and future operators of a domain, it is necessary that the representation be as precise as possible, i.e. clearly map to the real-world domain.

Therefore, web services must be semantically described (Medjahed and Bouguettaya, 2005). Service semantics include (1) models of services themselves and (2) models of the relationships between services and other web resources. The former is normally referred to as service ontology; an example is the OWL-S service ontology described in the following section. The latter model types aim at providing meaning to web services with reference mainly to the 'things' they require (inputs) and the 'things' they produce (outputs). Such models integrate the ontology of web services with the ontology of other web resources. The result is the creation of semantic web services instead of mere syntactic web services. This paper aims to provide a contribution in this area.

Currently languages, like the Web Services Description Language (WSDL), provide the syntactic means to describe web services (Paar, 2003), but lack in providing a semantic underpinning. In order to harvest all the benefits of web services technology, a framework has been developed for deriving business semantics from syntactic

descriptions of web services. The syntactic descriptions are interpreted in order to extract their real-world business content and then represent such content in domain/service ontological models. The benefits of such a framework are two-fold. Firstly, the framework provides a way to gradually construct domain ontologies from previously defined technical services. Secondly, the framework enables the migration of syntactically defined web services to semantic web services that are integrated within their respective domain ontologies. It should be noted that the proposed framework can in principle also be applied to the development of new web services or services derived from existing legacy system functionality. However, in this paper focus is given to the transformation of existing web services with limited or no semantic underpinning. Evidence of the significance of such a problem is provided by the general lack of semantic support to web service definitions by development technologies, such as .NET and J2EE, as well as by web service description languages, like WSDL. Moreover, as web service descriptions collected over two domains demonstrate (see Bell et al. (2005) and Section 6 of this paper), industrial web service development projects have tended to not model semantics.

The study follows a design research approach which (1) identifies the problem area and its relevance from an industrial case study and previous research, (2) develops the framework as a design artifact and (3) evaluates the application of the framework through a relevant scenario.

The paper is structured as follows. The following section will place the research within the context of the web services literature and previous related work. The relevance of the

problem of deriving semantic web services will be demonstrated. Afterwards the research approach adopted for this work will be presented, followed by a description of the proposed framework as well as the application of the framework to web services specifications drawn from an industrial project. The framework will then be evaluated and discussed.

2. Background

Web services have been an achievable distributed development alternative for a number of years, with a more recent focus on service composition. The Business Process Execution Language for Web Services (BPEL4WS) has proven to be the arena of this web services composition debate – evolving from IBM's Web Services Flow Language (WSFL) and Microsoft's XLANG. Issues such as semantics, expressiveness and adequacy have been raised (Staab et al., 2003). This is apparent in a typical service environment where requestor and provider matching require concept translation between parties. The knowledge transformation processes have had little if any coverage (Sycara et al., 2004). It is in recognition of these issues that review of the web services paradigm is undertaken before addressing some of the semantic approaches. The World Wide Web Consortium (W3C) Web Services Architecture Working Group defines web services as "a software application identified by a Universal Resource Identifier (URI), whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. Web services support direct interactions with other software agents using these XML-based messages exchanged via Internet-based protocols". The move to web services technology is driven by three communities: (1) The standards groups of the

W3C, the Organization for the Advancement of Structured Information Standards (OASIS) and the Global Grid Forum (GGF); (2) Middleware vendor adoption and; (3) The open-source community. Figure 1 summarizes the result of much of this activity in a layered model, highlighting dependency on the underlying service description in WSDL.

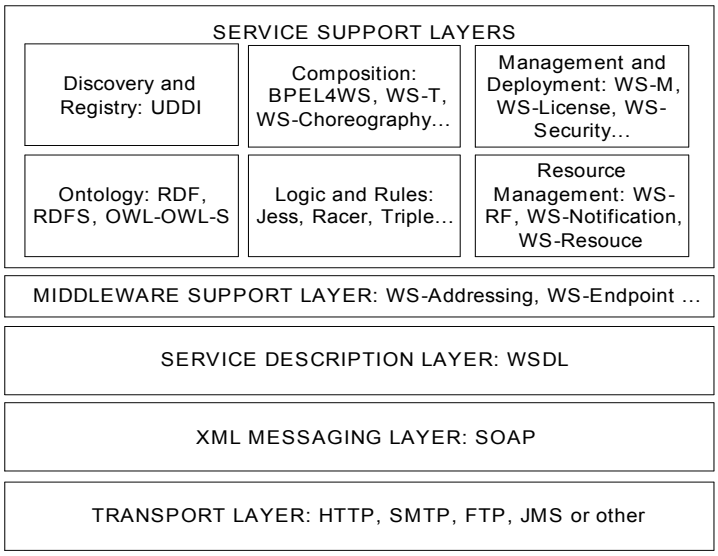


Figure 1: Web Service Technologies

Web services can be thought of as remote procedure calls over the web. The messaging is XML-based conforming to the Simple Object Access Protocol (SOAP). The form and structure of this service communication is described in another XML document (namely WSDL). The discovery of web services, typically carried out using a Universal Description, Discovery and Integration (UDDI) registry, provides a yellow page style lookup on available services. The approach relies on common business and service categorizations having utility across a community. The Semantic Web has added ontology to the web services stack; this layer is supported through the Resource Description Framework (RDF) and Schema (RDFS) languages as well as the Web

Ontology Language (OWL). These semantic languages have enabled the relation between web resources (including subclassing) to be made explicit.

XML and web services provide a low learning curve (Sheth and Miller, 2003) and resulting in a wider adoption. The amount of middleware and tooling gives the practitioner varied choice, but with limited clear guidance on how and when to use such technology. The diversity of approaches in WSDL creation typifies this. For examples of such approaches see (Paolucci et al., 2003; Kleijnen and Raju, 2003; Gronmo et al., 2004; Fremantle et al., 2002; Vinoski, 2003).

The Semantic Web is rooted in the Scientific American article from Berners-Lee et al. (2001, p.3) who state "The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in co-operation". The initial ideas were targeted toward static web pages and it is unsurprising that they gained resonance in the web services community. This transition can be seen in the often cited paper by Mcllraith et al. (2001, p.46) who describe semantic web services as making web services "computer-interpretable, use apparent, and agent-ready".

Current intersections between web services and the Semantic Web have delivered a diverse body of research. The agent community (Gibbins et al., 2003 ; Martin et al., 1999; McIlraith et al. 2001; Sycara et al., 2004; Wang et al., 2002) has recognized the benefit of ontology if computer-to-computer web architectures are to be achieved. Furthermore, the combination of service and domain ontology is seen as a key to achieving service synthesis (Chen et al., 2003). Work on service ontology is currently

centered on OWL-S, Web Service Modeling Ontology (WSMO) and WSDL-S (Akiraju et al., 2005; Lara et al., 2004) groups. Recognizing the progress, by the DAML Consortium and others, attention has moved from ontology languages to the application of services.

OWL-S is used as a basis for this work due to its maturity, although either WSMO or WSDL-S could also have been used. WSMO, as with OWL-S, provides a common semantic model for services, with additional focus on goals. WSDL-S offers a lighter approach enabling existing WSDL elements to reference OWL-based domain ontologies. All service ontology proposals have been submitted to the World Wide Web Consortium. The aim is to provide semantics to web services, either through common description in the case of OWL-S and WSMO or annotated relations to domain ontology with WSDL-S.

Others have identified the need for specialized common concepts within a web service context (Cardoso and Sheth, 2003; Curbera et al., 2002; Khalaf and Leymann, 2003; Paolucci and Sycara, 2003; Tosic, et al., 2002), with one example being quality of service. These concepts represent glue homogenizing a wealth of asymmetrically described web resources. New issues become pertinent in a semantic web of "great number of small ontological components consisting largely of pointers to each other" (Hendler, 2001; p.31). This semantic web service environment, with recognition of the need to combine service and domain ontology, warrants research that identifies practical approaches for businesses to combine the OWL-S Service ontology with existing or new domain ontology. The foremost question in semantic service orientation is how best this should be undertaken. The research described in this paper points in this direction.

3. Rationale and Research Design

The study follows a design research approach. Design research is a "search process to discover an effective solution to a problem" (Hevner et al., 2004). The relevance of the problem for the research community must be demonstrated and the solution must be effective to a satisfactory level. Effective solution may not (and generally does not) coincide with the "best" or "optimal" solution. The effectiveness of the solution must be demonstrable through an iterative evaluation of the designed artifact(s).

The design research presented in this paper is methodologically based on and adapted from the approach described by Nunamaker et al. (1991) and the guidelines presented by Hevner et al. (2004). The research outputs will also be described according to March and Smith's (1995) terminology for design research.

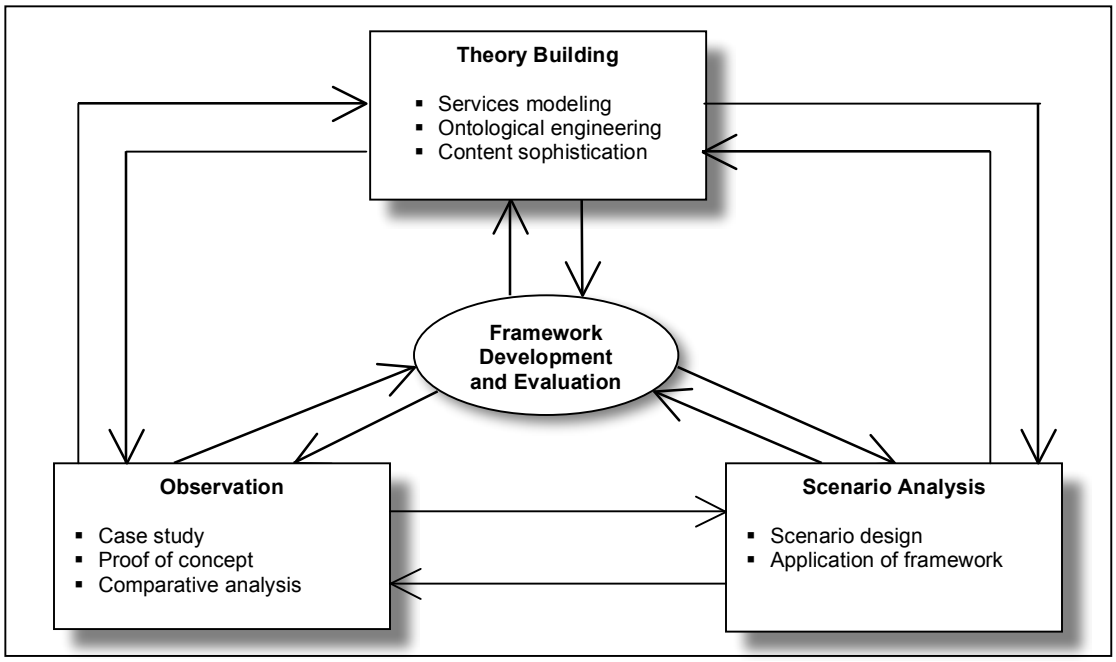


Figure 2: Framework development and research strategies (adapted from Nunamaker et al. (1991))

Adapting Nunamaker's multimethodological approach to design research (Nunamaker et al., 1991) the following four research strategies are adopted (Figure 2):

- Theory building: The study is theoretically based on previous work conducted in the areas of web services development and ontological modeling. The proposed framework builds upon this theory by covering an existing gap represented by a lack of integration between web services development and ontological modeling. Although the dominant literature concurs on the necessity of ontologically grounding web service descriptions, limited work has been carried out on the convergence of web service models and ontologies. More specifically the research builds on the previous work on interpretation of system models to derive ontological models.
- Scenario analysis: The developed framework has been evaluated primarily through its application to a typical web services development scenario. The scenario itself is not identifiable with a live project given the novelty of the research; however the scenario description and settings are drawn from a live industrial project described later in this paper. The application of the framework to the scenario represents the development of a "proof of concept" project whose outcomes are to be evaluated by the company from which the original service code is taken.
- Observation: Two sets of observations have been conducted. The first set of observations concerned a case study consisting of a live industrial project in which technical web services were developed. This case study was observed in

order to understand how a typical software development project currently organizes the development of a web service-based system. This observation allowed the research team to understand the limitations of services developed with current technology and methods in industry. The second set of observations was carried out on the proof of concept developed to evaluate the framework.

- Framework development and evaluation: The iterative development and evaluation of the framework for deriving semantics from syntactic or technical web services was the focus of this research.

The aforementioned strategies permeated the research as a whole. The strategies themselves should not to be considered as process steps, but rather as means of organizing the researchers' thought processes. All strategies were influential during every step of the study. In terms of the iterative cycle adopted to materialize the various research artifacts (i.e., constructs, models, method and instantiations), the steps that were followed are schematically outlined in Table 1.

Phases of research	Individual steps
Identify problem relevance	Conduct literature review Analyze industrial case study Identify gap(s)
Framework design	Define scope of framework Define underlying concepts and constructs Define framework process Define framework artifacts (input and output)
Framework evaluation	Apply framework to a realistic scenario Observe framework in action with proof of concept
Improve and re-evaluate framework	Identify limitations or areas of improvement Refine (re-design) framework (iterate previous two steps)
Communicate and discuss research	Identify limitations and further potential benefits Define directions for future work Disseminate (e.g., present and publish findings)

Table 1: The adopted design research process (based on guidelines by Hevner et al. (2004))

The instantiation of the research process is documented throughout this paper. The previous section highlighted an existing gap in the current literature. The following sections will provide further evidence of such a gap, confirming the relevance of the problem investigated. The results and artifacts of the phases and steps documented in Table 1 will be presented in a systematic manner.

4. Industrial Grounding

The premise to the research is based on the observations derived from an industrial case study project. The case study served three purposes:

- To demonstrate from a live large-scale web service-based project that the way in which web services are currently used mostly relates to the adoption of a technology which better enables good software engineering principles like modularization. In other words, the reasons that drove the use of web services were not the objectives of the Semantic Web, but good system design which does not necessarily require semantic content to be as explicitly represented as possible.
- To make the case for the need for a framework that extracts the semantics from syntactic web services and transforms that content into rich technology-agnostic representations. In fact, if the Semantic Web were to become mature, it would be necessary to cope with semantically improving previously developed web services.

- To design a realistic scenario to evaluate and refine the framework defined by this research. Due to the current level of immaturity of the Semantic Web and the skepticism of some (including researchers and practitioners), the framework could only be experimented on a simulated pilot-project. Such a project would require a scenario designed on the basis of previously developed services with the related experience and domain expertise.

S-Service is a medium-sized software development company located in southern Italy and founded 30 years ago. S-Service is specialized in the development of information systems for the healthcare sector and local public administrations (PA). Over the years it has extended its presence throughout the national territory becoming one of the leading software/information systems development firms for the healthcare and PA sectors.

This case study concerns a large development project carried out at a regional level for the Italian National Health Service (INHS). The project is aimed at the realization of a series of software services strategically intended to improve the quality of service of the INHS allowing medics, healthcare staff and citizens to directly interact with local and regional health structures through the Internet. The individual services were allocated to groups called Network Application Services (NAS). Ten NAS were defined and managed by distinct subprojects. The case study therefore refers to a coordinated project divided into ten subprojects. The NAS were from a technological perspective based on web services.

The aim of the project was to design a family of Information Technology (IT)-based services made available on a network (Internet and/or intranet) to operators of the Regional Health Service (Servizio Sanitario Regionale or SSR) in Puglia (Italy). Such operators include healthcare agents (for example, medics) and citizens (the customers of the SSR). The services developed extended pre-existing applications so as to make certain functionalities available online via distributed web service-based architectures. The web services developed are intended to provide added value to all agents by improving the level of direct interaction with the healthcare structures. The main areas of development concerned the administration of prescriptions and the provision of clinical and specialist healthcare services. These services together addressed two fundamental objectives: (1) to provide on request the complete clinical history of a patient and (2) to manage accounts (receivables and payments) between the healthcare structures involved. More specifically the functionalities that were developed include, for example:

- Retrieval of patient's personal and medical details
- Retrieval of patient's exemption status
- Registration of prescribed medication, specialist care and hospital admissions
- Scheduling of appointments for specialist services provided by the healthcare structures depending on their availability and the patient's needs
- Consultation of medical reports (e.g., diagnostic reports) by medics
- Retrieval of critical clinical information in emergencies
- Calculation of expenditures of the services provided

Web service technology was chosen because of its ability to expose, in the form of services, functionalities of previously developed applications with the least invasive intervention possible. This allowed the organizations involved to easily integrate such

services without greatly affecting the organizational structures and with minimal impact on their pre-existing information systems. Therefore, a web service based approach to carry out this transformation was thought to be more favorably accepted by the client.

Observations carried out on the projects described above led to the realization that the web service models and code were well developed from a software engineering perspective, hence sufficiently suited for the short and medium term objectives of both the developer and the client. In the long run however the lack of well-defined semantics underlying the web services would lead to their limited applicability in the future Semantic Web. A framework was therefore developed with the aim of alleviating the semantic deficiencies of the web services designed and implemented with current technologies such as WSDL, SOAP and J2EE. The framework was subsequently evaluated in a pilot project described in the following sections.

5. Framework for Deriving Ontological Models from Web Service Descriptions

5.1 Underlying philosophy and concepts

A framework has been developed for deriving semantic content from syntactic web services and representing such semantics in ontological models. The framework is based on the principles of Content Sophistication described by Partridge (1996) and Daga et al. (Daga et al., 2005). Content Sophistication represents a process for improving the semantic contents of legacy systems along several dimensions and representing such improvements in technology-agnostic conceptual models. The framework proposed in

this paper provides the basis for interpreting the semantics of syntactic web services in a similar fashion. In this case the syntactic web services can be viewed as “legacy code” from the perspective of the Semantic Web and its ideals. In fact in order to achieve the claimed benefits of the Semantic Web, it is necessary for web services to be semantically well defined and related to other types of web resources (Fensel and Lassila, 2000). In this sense it is not exaggerated to state that, for the Semantic Web, syntactic descriptions of services developed today represent the 'legacy of the future'.

According to the Collins Concise Dictionary (2001 edition) a framework is “a structural plan or basis of a project” and “a structure or frame supporting or containing something” (p.567). The principal artifact of this research can be considered a framework for the following reasons: (1) The framework defines a generic *structure* for a process of semantic interpretation and transformation of syntactic service content; (2) The framework is not a full-fledged methodology but *contains* a basic set of guidelines and heuristics, which can be integrated within software development or reengineering methodologies; and (3) The framework can be tailored to transform notational representations of web services defined in any language. The components of the framework will be expressed primarily in terms of activities and input/output artifacts.

At the heart of the framework is the adoption of ontology to drive the derivation of semantic content from syntactic web services. From a philosophical perspective ontology can be defined as a set of things whose existence is acknowledged by a particular theory or system (Honderich, 1995). Such ‘things’ include both types (such as the class of

Patients) and individual elements (such as the patient *John Smith*). The adoption of such a definition is important because, when compared with more computationally orientated definitions of ontology (for example, Gruber (1993; p.1) states that "an ontology is a specification of a conceptualization"), there is an explicit reference to a system's ontic commitment (i.e., things whose existence is acknowledged or recognized). This leads to representations that are more closely mapped to real world objects. Such mapping or reference (Frege, 1884) is essential to ontological modeling. The meaning of a sign, used, for example, to denote a service or a parameter, becomes well understood when it is possible to identify the thing(s) the sign refers to.

The focus of the framework presented in this section is the discovery of the semantics underlying a service description in its fundamental parts (mainly name and parameters). This process of discovery, called interpretation, identifies those real world objects that individual service parts ontologically commit to (or refer to). The semantics that are unraveled in this way are then represented in technology-agnostic domain and service ontology models.

Semantic web services require an ontological underpinning. High level service ontologies such as OWL-S are essential but not sufficient to exploit the full potential and the claimed benefits of the Semantic Web. Along with the technical means (e.g., programming tools such as J2EE and .NET, or description, discovery and messaging technologies such as WSDL, UDDI and SOAP), it is necessary to have a complete integration between knowledge bases (in the form of ontological domain models) and

functional offerings (in the form of semantic web services). In other words, web services must be described in relation to the classes and individuals modeled in shared and commonly agreed web ontologies. Currently the Web Ontology Language (OWL) and its predecessor DAML are the languages in which most ontological models are represented.

5.2 Scope of the Framework

The framework addresses the following objectives: (1) Derivation of semantics from previously developed web service syntactic descriptions; (2) Representation of the derived semantics in ontological models; and (3) Integration of models of semantic web services with models of other web resources. These objectives define the scope of the framework. A process was defined in order to achieve the objectives listed above. The process presented here is the final version of the iterative design research conducted.

It is important to state that the scope of the research expanded as the work progressed. This is not unusual given the iterative nature of design research. The initial scope of the project was limited to deriving semantic content from technical services and representing them in a commonly accepted service ontology such as OWL-S. However, as it became apparent, this led to models of services not totally integrated with models of other types of resources. Thus, the third objective emerged in recognition of the need for a common representation of all web resources in order to facilitate the discovery and composition of services. It is however beyond the scope of this paper to describe in detail how the

ontological models derived from the framework can be used by a semantic web search facility to discover and compose services.

5.3 Framework Process and Artifacts

The process, which drives the discovery and representation of semantic content from technical web services, is summarized in Table 2. The process is iterative and its outcome (defined in terms of ontological models) outlives one specific reengineering project. The framework's ongoing mission is to develop (within and across domains) interlinked ontological models for the Semantic Web. These models represent simultaneously all types of resources including service offerings. The process consists of three main activities: service interpretation, concept scoping and harmonization. These activities have been adapted from the Content Sophistication process presented by Daga et al. (2005). As a whole the process takes in technical service descriptions and produces ontological representations. The individual process activities also require and produce artifacts which progressively lead to achieving the ontological models.

Activities	Description	Input Artifacts	Output Artifacts
Service interpretation	A service description is broken down into its fundamental parts (e.g., name, input and output parameters). Each part is interpreted in order to represent its ontic commitment.	▪ Web service descriptions (e.g., WSDL code)	▪ Individual service ontic commitment models
Concept scoping	The concepts represented in the service ontic commitment models are either mapped to pre-existing ontologies or assigned to newly developed ones.	▪ Service ontic commitment models ▪ Domain ontologies	▪ Objects incorporated or mapped to ontological domain models
Harmonization	Services are represented within ontological models and related to other domain objects.	▪ Service ontic commitment models ▪ Domain ontologies	▪ Domain ontologies integrated with service representations

Table 2: Process for deriving semantic content from web services

5.3.1 Interpretation

The first activity is Service Interpretation. This activity works on service descriptions with limited or no explicit semantic underpinning. The descriptions are normally represented in the form of a service name with input and output parameters and termed Web Elements. The parameters themselves are named and typed. For example, in WSDL a typical service description can be found as a combination of service signatures and of complex data type definitions. The number of Web services reflects the number of operations defined within the WSDL description. The number of elements (*Web Service Elements*) identified are the service name, the parameter names and the output name (all resident in the WSDL document). The data types are described first, decomposing each type into named elements of base type such as String, Integer or Array. The service signatures follow, under a port and operation name in WSDL terminology. Each operation details the input and output message, parameters and return values, using the earlier type definitions.

Interpretation is aimed at representing the service's ontic commitment. This means unbundling and making as explicit as possible the real world (business) objects that the service descriptions recognize the existence of. In fact interpretation is defined as "the act of clarifying or explaining the meaning" of something (Collins Concise Dictionary 2001, p.761). Analogously identifying the real world objects that a service commits to is an act of clarifying the meaning of service descriptions. Interpretation achieves its best results

when actual service instantiations, in terms of actual data inputted and produced by the services, are made available. Individual level instance data can in fact greatly help in clarifying the meaning of type level data. However this is sometimes not possible due to privacy and confidentiality constraints. This was the case in the pilot project described in the following section.

Interpretation produces Service Ontic Commitment (SOC) models. SOC models adopt the Object paradigm (Partridge, 1996). The Object paradigm, not to be confused with the Object-Oriented paradigm, was specifically designed for business modeling and is quite effective in precisely representing real-world semantics. Precise representation, in this case, refers to being able to clearly identify the mappings between the representation and the represented. It is beyond the scope of this paper to describe the Object paradigm in detail. It is sufficient to note that this paradigm models all "things" (including classes, individuals and relationships) as objects with a four-dimensional extension. The paradigm is attribute-less unlike more traditional paradigms (e.g., entity-relationship or object-oriented).

5.3.2 Concept Scoping

Concept Scoping is aimed at allocating the "committed" objects of the SOC models to pre-existing ontological models or, in the case of a newly explored domain, to newly developed ontologies. There are various ways in which content scoping can occur. With

reference to an ontology language like OWL, new objects (such as classes, properties and individuals) can be incorporated into an ontology as exemplified in Table 3.

Ontologies in general should be shared across the community that they reference. The cause and implication in choosing to extend or develop ontological artifacts warrant coverage. Two reasons for creating a new ontology are ownership restriction and privacy concerns. The ownership of an ontology document on the Web may be outside the control of the engineer and result from an inability to further develop. A likely result is the creation of a hybrid ontology that references and extends the common ontology. In terms of privacy, detailed relationships to real world objects may expose competitive knowledge from within the organization. Hybrid ontology is again a likely approach with the shared ontology (or core ontology) being referenced by the organizations in specialized, private knowledge in a localized ontology.

Object Type	Method of Incorporation
Class	Define the class a newly developed ontology without any relation to pre-existing ontologies Define the class as a subclass of a class defined in a pre-existing ontology Define the class as an instance of a class defined in a pre-existing ontology Define the class as equivalent to a pre-existing class
Property types	Same as for classes
Individuals	Instantiate a class

Table 3: Methods of incorporating identified classes, properties and individuals

5.3.3 Harmonization

Web services are resources which provide agents (human or software) with business offerings whose instantiations produce real world effects. Web services can use other web resources and can produce new resources. In this sense services will become an integral part of the Semantic Web and as such should be modeled similarly and in relation to other types of web resources. Harmonization is aimed at overcoming the

traditional divide that is generally adopted between static and dynamic resources. The argument here is that if distinct types of representations are used for web services and other resource types, the necessary integration and semantic binding between them would become more difficult to resolve. Ontological models, which simultaneously represent all types of web resources, provide the benefit of facilitating the semantic discovery and composition of web services by software agents (Hendler, 2001). Agents would be able to traverse semantic graph lattices (or networks) in which services would be associated with the objects they use, transform and produce.

Harmonization uses the SOC models produced by Service Interpretation and the domain ontologies used in Concept Scoping to produce domain ontologies which incorporate service representations. The output artifact is represented in an ontology language such as OWL.

6. Framework Application

The framework was applied to a simulated project with the aim of testing, improving and evaluating the framework. Overall it was necessary to pragmatically validate the assumption that the principles, concepts and process underlying the framework were sufficiently well grounded to: (1) Achieve the intended objectives (and purposes) of the framework; and (2) Achieve such objectives in a setting that would presumably be, to the best of the researchers' knowledge, as industrially realistic as possible. It must be said that the researchers' backgrounds were mixed (academic and industrial) and two of the researchers were involved in the original project in which the WSDL code was produced.

As previously stated, due to the novelty of the research and to the current state of immaturity of the Semantic Web, the decision to carry out the validation of the framework through a "simulated project scenario" was considered to be the optimal framework evaluation strategy.

The service descriptions were coded in WSDL. The pilot project worked on a subset of the overall services produced by the original project. The research was carried out on five WSDL documents containing forty-five web services (or operations in WSDL terminology). Due to the limited length of the paper only one web service interpretation is shown as an example in this section. The objects that are derived from the service's interpretation are then scoped and harmonized.

The following worked example is carried out on a web service called *getGPClinics*. Given a specific healthcare region the web service determines which general practitioner clinics are open on the specified date. The following subsections will exemplify how the framework can be used to interpret, scope and harmonize *getGPClinics*.

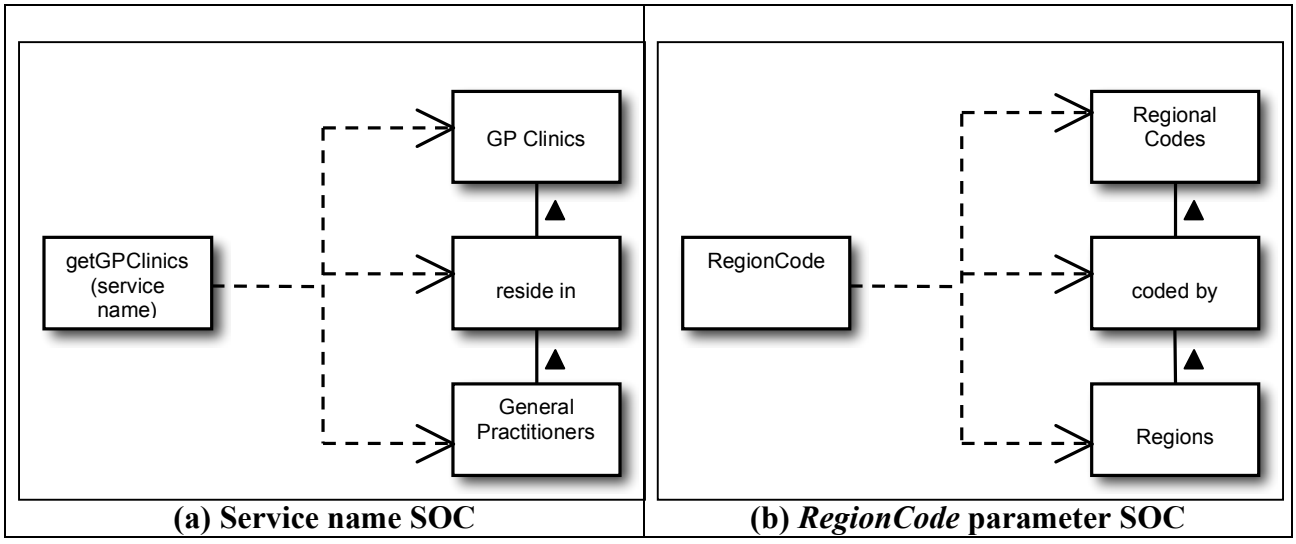
6.1 Interpretation

Table 4 summarizes the service's main elements that will be interpreted. These elements are the service's name, input and output parameters. All these elements ontologically commit to one or more real world objects. The service ontic commitment models are illustrated in Figure 3.

Service name: getGPClinics	
Description	Provides a list of General Practitioner Clinics open in a given region (or territory) on a specific date.
Input parameters	RegionCode: String RefDate: Datetime
Output parameters	GPClinics: Vector of String

Table 4: *getGPClinics* Web Service

As the diagrams of Figure 3 show, each part of a service can be unbundled and mapped to real world objects that clearly define the part's semantics. The object paradigm, as stated previously, help in this unbundling process given that all objects are explicitly revealed. For example, even the *reside in* relationship between *General Practitioners* and *GPClinics* is represented as a "committed" object. This type of representation is similar to OWL in which relationships are explicitly represented as properties.



(a) Service name SOC **(b) *RegionCode* parameter SOC**

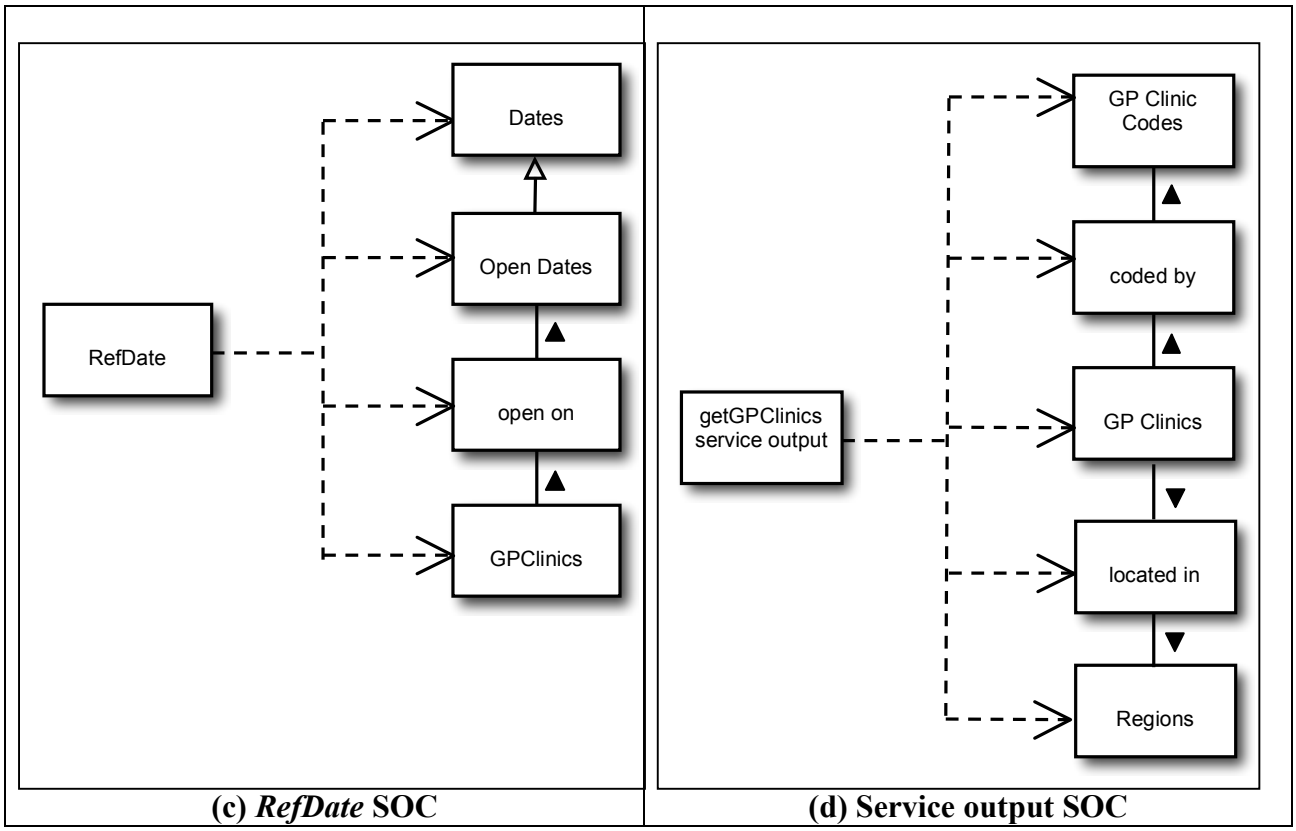


Figure 3: Service Ontic Commitment (SOC) models of the *getGPClinics* web service

6.2 Content Scoping

Content scoping uses the SOC models to create or refine the domain ontology that the overall process develops. In this case the domain ontology that is being developed from the web services is a healthcare ontology specific to the Italian National Health Service. The initial domain model is represented in Figure 4. It is an initial model because only one web service has been interpreted in the example presented in this paper. Further iterations with a range of services have refined the model. The objects identified in the previous activity and represented with their relationships in Figure 4 are scoped to either the newly developed ontology or to pre-existing ontologies. Table 5 summarizes the decisions and actions to be taken for each object. Tools to better identify existing (and

supported) ontologies help with scoping, including the opportunity to carry out semantic search and object visualization. These enable the engineer to identify related objects through subsumption or common neighbors. A range of tools exist to support the engineer, including, for example, Swoogle and SemWebCentral. Scenario analysis provides a practical influence as models are formed and tested with real-world usage scenarios.

Objects	Scoped to Ontology	Action Decided
Regional Codes	Geopolitical Regions (defined in previous legacy transformation projects (see (Daga et al., 2005))	Subclass
coded by		Subclass
Regions		Subclass
GP Clinics	Healthcare	Develop new ontology
reside in		
General Practitioners		
GP Clinic Codes		
coded by		
GP Clinics		
Open Dates		
open on		
GPClinics		
Dates	Time (defined in previous legacy transformation projects (see (Daga et al., 2005))	Define equivalence (e.g., OWL equivalent class which refers to the definition of synonymous classes)

Table 5: Scoping of *getGPClinics* SOC

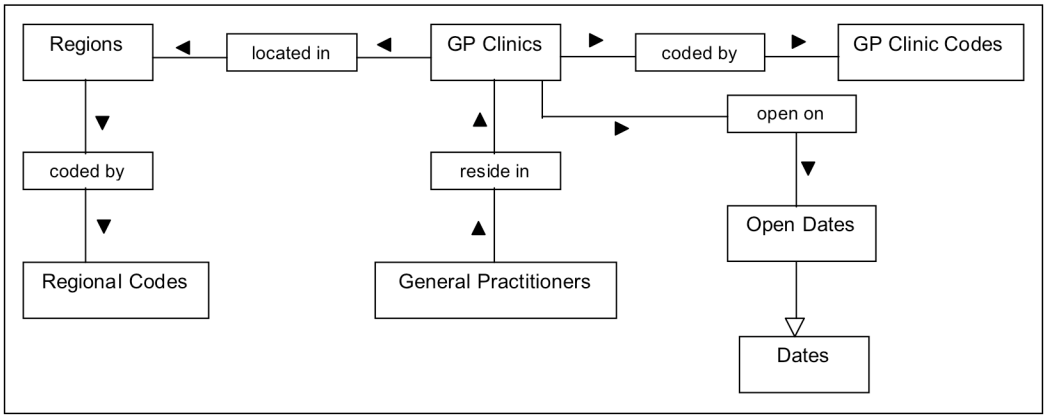


Figure 4: First-cut domain model

6.3 Harmonization

In harmonization the web service(s) is combined with the domain ontology. Ontologically this enables an explicit mapping between a service (with its parts) and the domain it serves. Figure 5 illustrates the harmonization model derived from the previous interpretation and content scoping. The *getGPClinics* service is defined within the OWL-S service ontology and its parameters are typed in relation to the respective domain ontology. The result is a combined semantic graph in which both domain objects and services are represented and linked together. Figure 6 exemplifies the mappings between the original WSDL code and the OWL-S/OWL ontology produced. Relationships with pre-existing ontologies are also highlighted.

7. Discussion and Evaluation

The research presented in this paper primarily targets the process and activities required to construct ontological domain and service models. This work acts as a counterbalance to the much larger body of work on the predominantly technical application of web services. Applying a design research approach to this area, with its support for process and constructed artifacts, has proved to be an auspicious choice. The approach enabled the development, validation and refinement of the framework. The holistic nature of the design approach has enabled practical issues associated with each step to be exposed. These issues are detailed in Table 6.

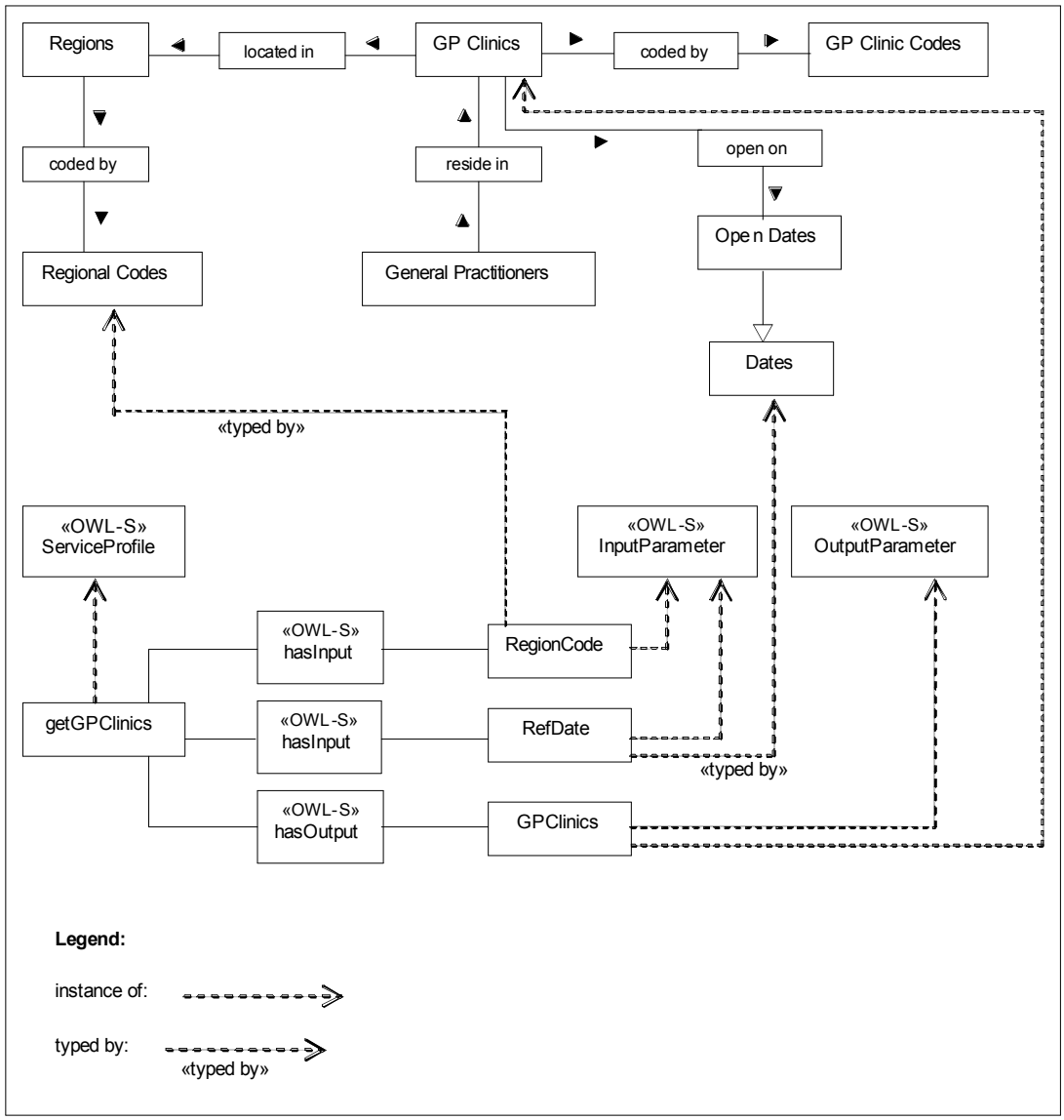


Figure 5: Harmonized model

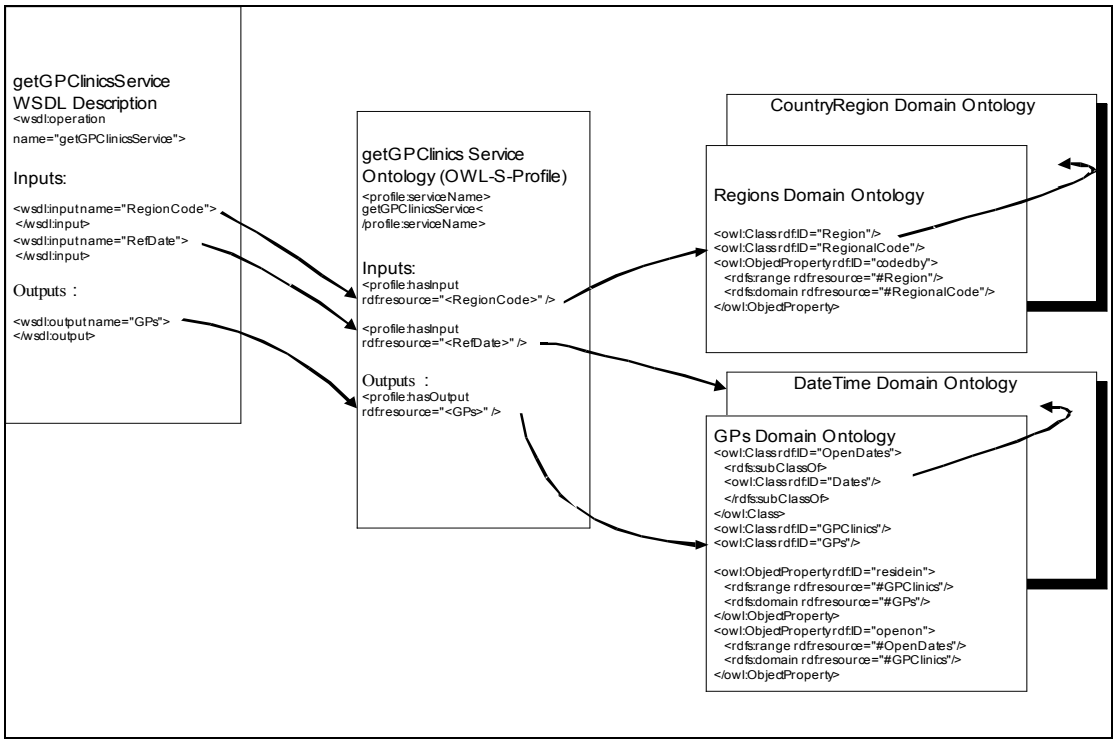


Figure 6: Service-Domain Ontology Topology

Framework Activity	Issues
Interpretation	▪ Ambiguous or misleading concepts may arise ▪ Over generalization of concepts ▪ WSDL type mismatches across the same concept ▪ The strategy used to create each web service will dictate the use of this validation. A service façade using base data types will yield less that a web service exposing richer business objects. ▪ Heterogeneous service inventories may require some concept normalization before decomposition ▪ Concepts often infer more that they make explicit, e.g. getGP may return a GP and his address. ▪ Without a knowledge of domain ontology the later normalization may become greater
Content Scoping	▪ Laziness motivates the creation of new, often duplicate, ontology when tooling is readily available ▪ The lack of a managed dictionary of available domain ontology hampers service selection and substitution.
Harmonization	▪ Without applying appropriate scenario analysis, the harmonization may result in an over rich, under used ontology.

Table 6: Framework Application Issues

The framework has implications on the important areas of service interaction or message exchanges, discovery of web services and service composition.

In terms of service interaction the ontological underpinnings of the models produced by the framework have the potential to improve the validation of service communication. Validated communication with a service endpoint is dependant on the type system in operation when starting such communication. Integrated domain and service ontologies will support improved validation through (a) the use of available ontological models, (b) richer types (or classes) and (c) the inclusion of type instances (for example, a region code parameter would be validated not only against type, but also against actual region codes). This improvement is reliant on the utilized service models including appropriate domain ontologies. It is the framework process activities that support the choice and form surrounding each domain model.

Discovery of semantic web services requires mechanisms that go beyond the syntactic search of UDDI or WSDL documents. The use of the OWL-S profile to bind together service and domain ontologies allows standard semantic searching by traveling throughout concept branches of particular domain ontological models. The semantic search over several models grounded in real-world "things" provides a greater scope for matching to a requestor's concept. Service discovery is fundamentally linked to service composition.

Due to its contribution in improving service knowledge, the framework supports the web service composition process in two areas. Firstly, assuming a workflow-based approach, a composition design tool would acquire and build on the discovery benefits already mentioned. It can be assumed that the greater semantic expressiveness of the ontological models would provide benefits by enabling access either to a larger or more appropriate group of available services (determined by the span of the concept tree or lattice traversal). Secondly, the same discovery benefits allow the workflow tool to describe a conceptual service that is then discovered dynamically at execution time. The latter approach is adopted by the grid community (Blythe et al., 2003).

To conclude this section, it is useful to discuss the implications to both the semantic web service community and its service transformation tooling groups. The semantic web service community has started to move on from a focus on pure knowledge to include rule and logic languages (e.g. Semantic Web Rule Language SWRL). This paper posits that this transition is premature without a clearer understanding of service knowledge through real world grounding. Ambiguity around the coarseness of web services is a symptom, implying a need for prescriptive approaches to service-orientation. The framework presented in this paper recognizes and addresses the need to apply rigor and repeatability to the ontology engineering process. Support of greenfield development through models derived from legacy systems is realized within the harmonization phase where linkages to existing domain ontologies are achieved. In a greenfield environment, the framework supports (a) the use of community ontological models, (b) the specialization of community models with additional concepts or (c) the creation of new

domain ontologies. Only with such focus on service knowledge will semantic web tools deliver benefits to the enterprise. Implied in the adoption of such an approach is the access to domain knowledge and experience that are able to unravel higher level business objects.

One area of tool support is the transformation of physical service descriptions, such as WSDL, into a semantically richer form. This process has tended toward simplistic, automated methods. The consequence, when coupled with code to WSDL transformation of earlier engineering phases, has been an inclination towards purely technical service models. For example, transforming WSDL to OWL-S has provided the practitioner with only a skeletal service description. Ad-hoc additional description, without consideration of what is being described or why it is being described, increases the risk of inadequate or unusable ontologies. Success is then only achieved through the tacit knowledge of the practitioner being applied during post transformation descriptions. The implication of the framework to this automated transformation activity is that: (1) Tool support will be required to use and manage a catalogue of ontologies when engineering WSDL or OWL-S descriptions; (2) Algorithms are required to dynamically scope and select such catalogues and; (3) Guidelines mapping tools to the ontology engineering framework should be in place prior to development. The proposed framework provides a basis for tool selection and strategy.

Conclusion

The research presented in this paper was aimed at resolving a problem related to the semantic expressiveness of web services developed primarily for the realization of

software systems that respect good design principles and meeting both functional and non-functional requirements. Web services however are web resources and play (or will play) a key role in offering distributed functionality in the Semantic Web. Notwithstanding the engineering and quality strengths that web service-based systems can possess, there is an area of concern related to the semantic expressiveness of the web services developed and their role in the future Semantic Web. The majority of web service-based projects today are conducted from a purely technical perspective. Current languages for implementing and describing services lack in providing the necessary level of semantic representation of the services themselves and of the domain objects that are subjected to the services' behavior. The case study presented provided evidence of these claims.

The problem just described was analyzed in order to design a solution in the form of a framework. The framework, based on a philosophy and concepts that place semantics at the heart of web services, was presented. At the heart of the framework is a process and related artifacts for interpreting, scoping and harmonizing the content of syntactically defined web services. The framework was developed through a design research approach and validated through its application to a scenario derived from a previous industrial large-scale project. The application of the framework demonstrated the framework's ability to develop ontological models which represent simultaneously both domain and service concepts. The implications of such a framework were discussed along with possible improvements and future work.

References

Akkiraju, R., Farell, J., Nagarajan, M., Sheth, A. and Verma, K., 2005. Web Service Semantics - WSDL-S, W3C, ed. In: *Frameworks for Semantics in Web Services,* 05/6 2005.

Bell, D., de Cesare, S., Lycett, M. (2005). Semantic Transformation of Web Services. In ITM 2005 – SWWS 2005 (Vol. 3662, pp. 856-865.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American, 284*(5), 34-43. SPRINGER-VERLAG BERLIN.

Burstein, M., Bussler, C., Zaremba, M., Finin, T., Huhns, M.N., Paolucci, M., Sheth, A.P. and Williams, S., 2004. A Semantic Web Services Architecture. *Internet Computing, IEEE,* **9**(5), pp. 72-81.

Blythe, J., Deelman, E., Gil, Y., Kesselman, C., Mehta, G., Vahi, K., et al. (2003). Mapping Abstract Complex Workflows onto Grid Environments. *Journal of Grid Computing, 1*(1), 9-23.

Cardoso, J., & Sheth, A. (2003). Semantic e-workflow composition. *Journal of Intelligent Information Systems, 21*(3), 191-225.

Chen, L. M., Shadbolt, N. R., Goble, C., Tao, F., Cox, S. J., Puleston, C., et al. (2003). Towards a knowledge-based approach to semantic service composition. In *Semantic Web - Iswc 2003* (Vol. 2870, pp. 319-334). Berlin: SPRINGER-VERLAG BERLIN.

Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., & Weerawarana, S. (2002). Unraveling the Web services Web - An introduction to SOAP, WSDL, and UDDI. *Ieee Internet Computing, 6*(2), 86-93.

Daga, A., de Cesare, S., Lycett, M., & Partridge, C. (2005). An Ontological Approach for Recovering Legacy Business Content. In *Proccedings of the 38th Hawaii International Conference on System Sciences*. Los Alamitos, CA: IEEE Computer Society.

Fensel, D., & Lassila, O. (2000). The semantic web and its languages. *Intelligent Systems and Their Applications, IEEE [see also IEEE Intelligent Systems], 15*(6), 67-73.

Frege, G. (1884). *The foundation of Arithmetic: A logico-mathematical enquiry into the concept of number*.

Fremantle, P., Weerawarana, S., & Khalaf, R. (2002). Enterprise services *Commun. ACM 45* (10), 77-82

Gibbins, N., Harris, S., & Shadbolt, N. (2003). Agent-based semantic web services In *Proceedings of the 12th international conference on World Wide Web* (pp. 710-717). Budapest, Hungary ACM Press.

Gronmo, R., Skogan, D., Solheim, I., & Oldevik, J. (2004). *Model-driven Web services development.* Paper presented at the IEEE International Conference on e-Technology, e-Commerce and e-Service, 2004.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition, 5*(2), 199-220.

Hendler, J. (2001). Agents and the Semantic Web. *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications], 16*(2), 30-37.

Hevner, A., March, S., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly, 28*(1).

Honderich, T. (1995). *Oxford Companion to Philosophy*. Oxford: Oxford University Press.

Khalaf, R., & Leymann, F. (2003). On web services aggregation. In *Technologies for E-Services, Proceedings* (Vol. 2819, pp. 1-13). Berlin: SPRINGER-VERLAG BERLIN.

Kleijnen, S., & Raju, S. (2003). An Open Web Services Architecture *Queue 1* (1), 38-46
Lara, R., Roman, D., Polleres, A. and Fensel, D., 2004. A Conceptual Comparison of WSMO and OWL-S, *Web Services: European Conference, ECOWS 2004,* September 2004, pp254-269.

March, S., & Smith, G. (1995). Design and Natural Science Research on Information Technology. *Decision Support Systems, 15*, 251-266.

Martin, D., Cheyer, A. J., & Moran, D. B. (1999). The Open Agent Architecture: A Framework for Building distribted Software Systems. *Applied Artificial Intelligence, 13*(1-2), 91-128.

McIlraith, S. A., Son, T. C., & Zeng, H. L. Semantic Web services. *IEEE INTELLIGENT SYSTEMS & THEIR APPLICATIONS, 16*(2), p46-53.

Medjahed, B., & Bouguettaya, A. (2005). A Dynamic Foundational Architecture for Semantic Web Services *Distrib. Parallel Databases 17* (2), 179-206

Nunamaker, J., Chen, M., & Purdin, T. (1991). System Development in Information Systems Research. *Journal of Managemennt Information Systems, 7*(3), 89-106.

Paar, A. (2003). Semantic software engineering tools In *Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications* (pp. 90-91). Anaheim, CA, USA ACM Press.

Paolucci, M., Srinivasan, N., Sycara, K., & Nishimura, T. (2003). *Towards a Semantic Choreography of Web Services: From WSDL to DAML-S.* Paper presented at the ICWS03, Las Vegas, USA.

Paolucci, M., & Sycara, K. (2003). Autonomous semantic web services. *Ieee Internet Computing, 7*(5), 34-41.

Partridge, C. (1996). *Business Objects: Re-Engineering for Reuse*. Oxford: Butterworth-Heinemann.

Sheth, A., & Miller, J. A. (2003). Web services: Technical evolution yet practical revolution? *Ieee Intelligent Systems, 18*(1), 78-80.

Staab, S., van der Aalst, W., Benjamins, V. R., Sheth, A., Miller, J. A., Bussler, C., et al. (2003). Web services: been there, done that? *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications], 18*(1), 72-85.

Sycara, K., Paolucci, M., Soudry, J., & Srinivasan, N. (2004). Dynamic discovery and coordination of agent-based semantic Web services. *Internet Computing, IEEE, 8*(3), 66-73.

Tosic, V., Esfandiari, B., Pagurek, B., & Patel, K. (2002). On requirements for ontologies in management of Web Services. In *Web Services, E-Business, and the Semantic Web* (Vol. 2512, pp. 237-247). Berlin: SPRINGER-VERLAG BERLIN.

Vinoski, S. (2003). It's just a mapping problem [computer application adaptation]. *Internet Computing, IEEE, 7*(3), 88-90.

Wang, F. J., Zhao, Z. F., & Han, Y. B. (2002). A dynamic matching and binding mechanism for business service integration. In *Engineering and Deployment of Cooperative Information Systems, Proceedings* (Vol. 2480, pp. 168-179). Berlin: SPRINGER-VERLAG BERLIN.