Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2009), *23*, 373–390. Printed in the USA.

doi:10.1017/S0890060409990060

Problem solving methods in a global networked age

JOHN DOMINGUE[1] AND DIETER FENSEL[2]

[1]Knowledge Media Institute, The Open University, Milton Keynes, United Kingdom
[2]Semantic Technology Institute Innsbruck, University of Innsbruck, Innsbruck, Austria

(Received December 1, 2008; Accepted January 15, 2009)

Abstract

We believe that the future for problem solving method (PSM) derived work is very promising. In short, PSMs provide a solid foundation for creating a semantic layer supporting planetary-scale networks. Moreover, within a world-scale network where billions services are used and created by billions of parties in *ad hoc* dynamic fashion we believe that PSM-based mechanisms provide the only viable approach to dealing the sheer scale systematically. Our current experiments in this area are based upon a generic ontology for describing Web services derived from earlier work on PSMs. We outline how platforms based on our ontology can support large-scale networked interactivity in three main areas. Within a large European project we are able to map business level process descriptions to semantic Web service descriptions, to enable business experts to manage and use enterprise processes running in corporate information technology systems. Although highly successful, Web service-based applications predominately run behind corporate firewalls and are far less pervasive on the general Web. Within a second large European project we are extending our semantic service work using the principles underlying the Web and Web 2.0 to transform the Web from a Web of data to one where services are managed and used at large scale. Significant initiatives are now underway in North America, Asia, and Europe to design a new Internet using a "clean-slate" approach to fulfill the demands created by new modes of use and the additional 3 billion users linked to mobile phones. Our investigations within the European-based Future Internet program indicate that a significant opportunity exists for our PSM-derived work to address the key challenges currently identified: scalability, trust, interoperability, pervasive usability, and mobility. We outline one PSM-derived approach as an exemplar.

Keywords: Future Internet; Large-Scale Interoperability; Semantic Web; Semantic Web Services; Web Services Modeling Ontology

1. INTRODUCTION

The intellectual contributions from the problem solving methods (PSMs) research field is largely judged to have been significant. However, outside of a core community the work is largely unknown, raising a "So what?" question. In short, what will the lasting legacy and impact of PSMs be? One interesting comparison that can be made is to contrast the fields of PSMs with that of ontologies. As a core component of the Semantic Web stack, research associated with ontologies is currently very strong, with a number of specific events (e.g., Formal Ontology in Information System Conference), a large number of publications each year, and a number of researchers and research groups who identify with the area as their main topic. The same cannot be said for PSMs.

However, we believe that the future for PSMs and PSM-derived work is promising. In the same manner that a good fit exists between the needs as found in the Web and the affordances of ontologies, we argue how a universe comprising *world-scale networked interactivity* can benefit greatly from PSMs.

The Web has proven to be an unprecedented success largely for facilitating the publishing and use of data on a planetary scale. We believe, however, that a paradigm shift is underway from a data-centric world to one of networked interactivity. Two significant changes are behind this shift. First, Web service technologies, which encapsulate computation behind an Internet accessible interface, are moving from enterprise specific solutions encountered only behind corporate firewalls, to widely accessible online resources. For example, Amazon now offers a range of services that provide on-demand access to their backend infrastructures. Using Amazon services one can store arbitrarily large amounts of data, operate a 1-TB database, run message queues of any size, and request computing capacity in a flexible dynamic fashion. Moreover, utility-based computing, leveraging Web service technology, changes the relationship between information technology (IT) and business. For example, GigoVox Media,

Reprint requests to: John Domingue, Knowledge Media Institute, The Open University, Walton Hall, Milton Keynes MK7 6KA, UK. E-mail: j.b.domingue@open.ac.uk

a startup podcast company, set up their entire IT infrastructure on top of Amazon's services for less than \$100 (http://www.amazon.com/gp/browse.html?node=341907011).

Second, the other major driver is the addition of mobile devices, primarily mobile phones, on the Internet. Today the Internet has 1.4 billion users (http://www.internetworldstats.com/stats.htm), whereas there are currently 3.3 billion mobile phone users with an annual growth rate of 22% (http://www.chinapost.com.tw/business/global%20markets/2008/05/26/158188/mobile-phone.htm). The convergence of mobile networks and the Internet will mean that the requirements for mobile device users will drive the shape of the Internet. From an Internet perspective mobile devices can be regarded as thin clients that store relatively little data and rely on interaction with remote services to provide user functionality.

Aligned with the shift from a data-driven network to an interactive infrastructure is that the sheer scale of data available on the Internet will necessitate the use of semantics. For example, in 2006, 161 exabytes (10^8 TB) of information was created or replicated worldwide. IDC estimates a sixfold growth in this metric by 2010 to 988 exabytes (a zetabyte)/year (http://www.azulsystems.com/products/analyst_report/idc_vendor_profile_competitive_edge.pdf). More generically, estimates are that although currently new technical information doubles every 2 years, by 2010 this will be every 72 hr. From a user perspective a similar story of large numbers and significant growth exists. In April 2007, MySpace had over 200 million registered users, making the MySpace population the fifth largest country in the world. Similarly, the 4 billion devices capable of producing data is set to grow by 50% by 2010.

Our view is that PSMs provide a solid foundation for creating a semantic layer for interaction at planetary scale. Moreover, within a world-scale network where billions of services are used and created by billions of parties in *ad hoc* dynamic fashion we believe that PSM-based mechanisms provide a very promising approach to dealing with the sheer scale systematically.

In the following section we give a brief overview of the history and legacy of the initial research carried out on PSMs. Sections 3 and 4 outline how PSM-derived frameworks have been applied to Web services. In particular, in Section 3 we first explain how PSMs were directly applied to Web services, and then Section 4 describes the Web Services Modeling Ontology (WSMO). The future of PSMs is covered in Section 5, outlining how PSM-derived technologies can be applied in the areas of Enterprise Computing, creating a Web of Services and supporting a Future Internet. The final section presents a number of conclusions.

2. HISTORY AND LEGACY OF PSMs

Work within the late 1970s on Expert Systems led to a number of significant successes and the promise of a world where highly valued expertise would become widely and cheaply accessible. A number of generic expert system shells emerged (e.g., KEE, ProKappa, CLIPS) that combined frames and forward chaining rules as a representation medium.

A little while after the early successes, however, a number of problems emerged that were related to the lack of inherent structure endemic within rule systems and the gap between the knowledge to be represented and the representation language. Large rule systems, such as R1 (McDermott, 1980), required significant maintenance teams and were difficult to reuse.

There was recognition that a paradigm shift was required. Through a thorough analysis of a number of rule-based (e.g., MYCIN; Shortliffe, 1974), frame-based, and Lisp-based systems, Clancey (1983) concluded that rules presented a too uniformly and weakly structured a representation for expert knowledge. He found that rules could be categorized according the roles they played in solving a given problem. Making structural, strategic, and support knowledge explicit was necessary to make the underlying system understandable, maintainable, and reusable.

Clancey (1985) introduced the heuristic classification method as a way to analyze and structure expert systems with an intermediate representation. In many senses heuristic classification is considered to be an iconic PSM in that it was the first and is the most widely referred to.

As can be seen in Figure 1, heuristic classification is composed of three basic steps:

1. *Abstraction*: Data related to a problem that needs to be solved are abstracted to a (usually preexisting) representation.
2. *Heuristic match*: Heuristic knowledge is used to identify relevant *abstract solutions* from the abstracted data.
3. *Refinement*: The identified abstract solutions are refined to a specific solution that represents the "answer" to the original problem.

Heuristic classification has proved to be a powerful technique, and has been used as the backbone structure for applications in the areas of medical diagnosis, lift design, and fault diagnosis in engineering artifacts.

Another key publication related to PSMs was the seminal paper by Allen Newell (1982) that outlined the need for a knowledge level above and distinct from the symbol level. This work formed a basis for several initiatives in the knowledge modeling area.

In the 1980s and 1990s a number of prominent initiatives began in Europe, for example, Knowledge Acquisition and Document Structuring (KADS; Wielinga et al., 1992) and later CommonKADS (Schreiber et al., 1999). The main legacy

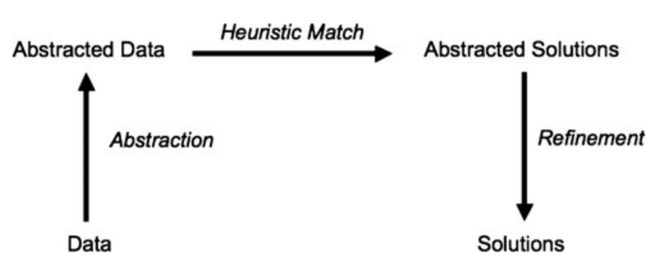

Fig. 1. The heuristic classification PSM.

from KADS and CommonKADS was the prominence given to libraries of reusable models capturing a number of orthogonal dimensions related to the creation and management of knowledge systems. From a PSM perspective the interesting model within the KADS framework was the *expertise model* that comprised three parts:

1. The *domain layer* contains the domain knowledge relevant to the system under construction. The domain layer folded together the conceptual structure for the domain (ontology) and the specific instance data.
2. The *inference layer* represented PSMs within an *inference structure* that comprised *inference actions*, which specified atomic steps in an inference process, and *roles* an abstraction over a domain specific input or output type of an inference action.
3. The *task layer* describes how the tasks are decomposed into subtasks and also providing a control and data flow structuring over the tasks.

The Intelligent Brokering on the World Wide Web (IBROW3) project (Benjamins et al., 1998) was the first initiative to realize the link between the newly emergent Web and PSMs. The goal of the project was to enable the on-the-fly creation of knowledge-based applications from reusable knowledge components found on the Web. The main result of the IBROW project was the Unified Problem-Solving Modeling Method Development Language (UPML) framework (Fensel & Motta, 2001) that distinguishes between the following classes of components:

- *Domain models*: Similar to the KADS domain layer, domain models describe the domain of an application (e.g., vehicles or a medical disease).
- *Task models*: Task models in UPML were structured differently than KADS, specifying the input and output types, the task to be achieved, and applicable preconditions. Tasks could be high-level generic descriptions of complex classes of applications, such as classification or scheduling, as well as more "mundane" problem specifications, such as currency exchange rate conversion.
- *PSMs*: Similar to tasks, PSMs specify input and output types and applicable preconditions. PSMs also specified a control and data flow over the subtasks and a postcondition satisfied once the PSM is executed.
- *Bridges*: Bridges made explicit the mappings between the different types of models to generate a coherent knowledge system. For example, bridges could contain mappings between the input roles of a task and a class within a domain model.

Each model type above was supported by an appropriate ontology. As can be seen in Figure 2, within UPML it was envisaged that a complete application would also require additional application-specific task and PSM knowledge.

Tool support was made available for the UPML framework through the Internet Reasoning Service (IRS; Crubezy et al., 2003), which provided semiautomated support to knowledge engineers in creating the bridges between the models. For example, automatically mapping a domain taxonomy to an input role of a PSM. In fact, two implementations were created of the IRS: one (IRS-Protégé) an extension of Protégé-2000 (Musen et al., 2000) and another (IRS-KMi) an implementation from The Open University based on the Operational Conceptual Modeling Language (Motta, 1999).

Although the IBROW3 project was visionary in its goal of bringing together the results from PSM-related research and the new possibilities provided by the emerging Web, the project was still situated within a world of knowledge modeling with its associated assumptions: all systems are encoded within a single knowledge representation language, for the most part according to a coherent ontology (ontology mapping was not seriously considered or addressed within IBROW3), and all reasoning occurred within in a single box. In the following section we motivate the need to rethink PSMs within the context created by an ever growing Web and Internet.

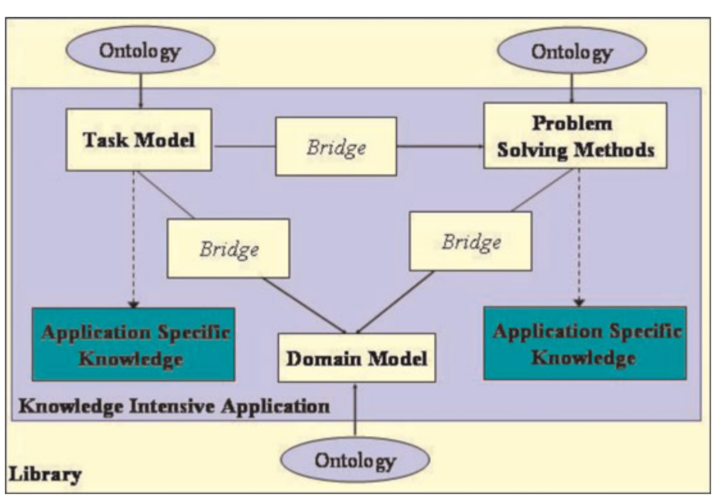

Fig. 2. The UPML framework. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

3. THE NETWORKED AGE: PROBLEMS AND SOLUTIONS

From an IT perspective today's context is best captured by the phrase that we live in a "networked age." Network technologies are the fabric through which we now communicate, work, and socialize. Indeed, networks are now seen as a driver for the economy, and a mechanism that shapes how we socialize and interact generally.

However, as mentioned, our new technological infrastructure brings challenges associated with the sheer volume of data and users. A key additional aspect tied to the extremely large growth is that the acceleration is driven by users and not by business. This means that unless a paradigm shift occurs the majority of the data available and communicated across our networks will have virtually no structure.

The grand challenge derived from the above is how to interweave highly dynamic data that is distributed across a mixed assortment of hardware and software platforms, owned by many parties, at Internet scale in a way that provides added value. Facilitating interoperability for extremely large, open, heterogeneous, and distributed environments coping with dynamicity is the computing challenge of our time in our opinion.

We feel two key technical solutions are necessary to address the above: semantics and services. Only machine-readable and processable descriptions of data and software components facilitates interoperability in a fashion that does not require human labor to process or to manually hardwire integration solutions. Moreover, solutions can become quickly outdated and are not reusable in a slightly different context. In consequence, these hardwired *ad hoc* integrations generate an additional burden for the next integration wave, that is, an exponentially growing problem. That is, scalable interoperability not only requires semantics, but it also cannot even be imagined without the usage of semantics.

The service concept addresses interoperability in four main ways:

1. *Service interface abstraction*: A service provides a layer, the service interface, which abstracts from the underlying software and hardware platform.
2. *Standard invocation-at-distance mechanism*: Services can be invoked over the Internet using standard protocols based on XML.
3. *Focus is on function*: By placing the stress on function, the volume and type of data can be dramatically reduced as only the public data that are the input and output of a service need be considered.
4. *Web service equates to a business service*: A Web service can stand as a computational proxy for a business service. That is, a business can directly offer microfunctionalities that provide value to a customer, for example, the booking of flights. This significantly raises the level of discourse available through online communication.

Figure 3 shows the typical use scenario for a Web service. Textual descriptions describing the Web service are placed within an online store called the Universal Description Discovery and Integration (UDDI) registry (2003). The UDDI registry additionally contains the Universal Resource Identifier (URI) of a XML-based description of the input and output types of the service: a Web Service Description Language (WSDL) file (WSDL, 2001), and the URI of the service end point. As can be seen in Figure 3, two roles are taken in the scenario. A *service consumer* finds an appropriate service by searching the UDDI registry (either manually or using string matching) and then invokes the service, using the service end point, by sending a Simple Object Access Protocol (SOAP) message (SOAP, 2003). *Service providers*, typically organizations, are responsible for making Web services available.

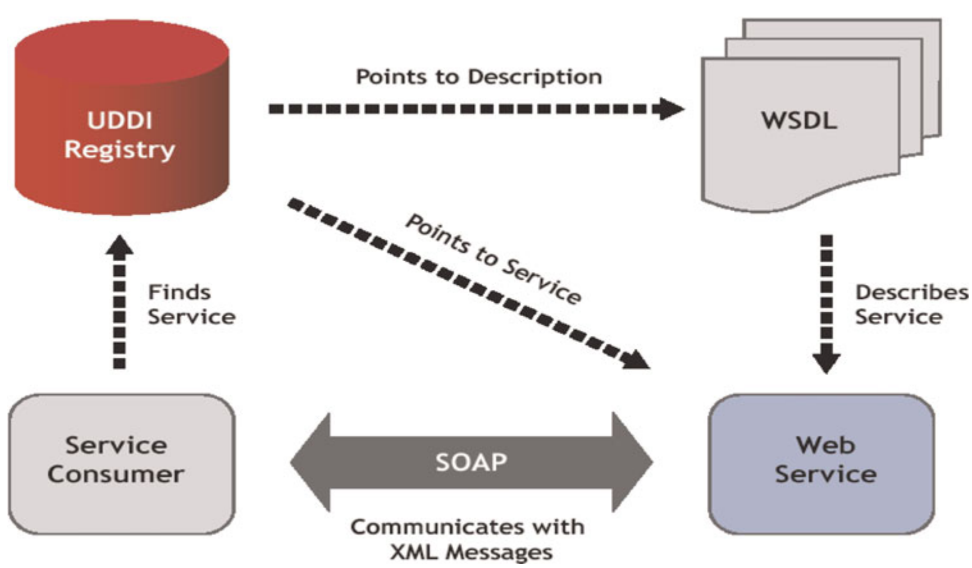

Fig. 3. The Web services usage scenario. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

As a technology for dealing with interoperability within enterprises Web services have proven very popular within the commercial sector and service-oriented frameworks (e.g., SAP's NetWeaver, http://www.sap.com/platform/netweaver/) now dominate the enterprise software market. Moreover, it is expected that service-oriented computing and service-oriented architectures will become the dominant application development paradigm within a few years.

A number of problems exist, though, with the current standards for Web services. The existing descriptions (e.g., WSDL) are impoverished in two ways. First, the descriptions are minimalistic in that they only describe the inputs and outputs of a service and not the service's capability. Second, because the descriptions are purely syntactic, the interpretation of the associated documents requires a (skilled) human that leads to a problem of scalability. Namely, all of the tasks associated with the creation of applications from online Web service components have to be carried out manually by IT specialists. Semantics, and in particular PSMs, provide a way to create enhanced descriptions of services alleviating the problems mentioned above.

3.1. IRS-II

IRS-II (Motta et al., 2003) was a semantic broker that mediated between task requests of users or clients and services available on the Internet. Within IRS-II the key insight was to link PSMs to external executable components through a number of *publishing platforms.* A publishing platform provided a wrapper for a component, making it available through HTTP-based requests. The execution of a "published" PSM resulted in a SOAP message sent to an external publisher and the relevant component code invoked. As can be seen in Figure 4 within IRS-II publishing platforms were created for standalone Lisp and Java code as well as Web applications available as an HTTP GET request and standard Web services with an associated WSDL file.

IRS-II provided a simple interface for publishing a PSM where the user simply had to specify the location of the IRS-II server, the name and home ontology of the target PSM, and the location of the component within a relevant publishing platform. Figure 5 shows an example of the interface for the Web service publishing platform.

We consider that IRS-II played a key role in bringing the power of PSMs to bear on Web services. However, IRS-II represented an incremental change, whereas what was required was a more fundamental reflection on how the basic principles underlying PSMs could be utilized in the Web service context from the ground up. In particular, we needed to take into account that the Web is an open, planetary-scale distributed environment, based on a universal naming scheme and a set of standardized protocols.

4. WEB SERVICES MODELING FRAMEWORK (WSMF)

The first radical rethink was WSMF (Fensel et al., 2002), which provided an overall framework for describing Web services. WSMF introduced two principles for semantic descriptions of Web services:

1. *Strict decoupling*: The various components that realize an application are described in a unitary fashion without regard to the neighboring components. This enables components to be easily linked and promotes scalability. This principle follows the open and distributed nature of the Web, and is similar to the no-structure-in-function principle outlined in de Kleer and Brown (1984).
2. *Centrality of mediation*: Building on the concept of bridges within the UPML framework, WSMF includes the notion of mediators to deal with mismatches that may occur between components. Heterogeneity can occur in terms of data, underlying ontology, protocol, or

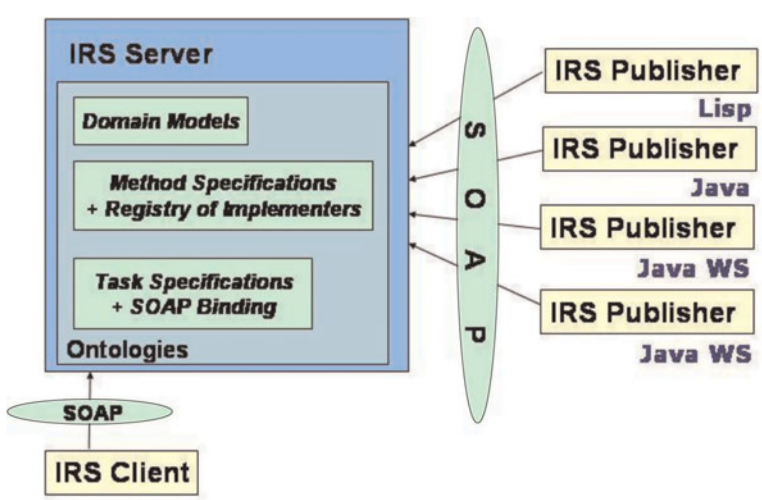

Fig. 4. The architecture of IRS-II. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

Fig. 5. The interface for the IRS-II Web service publisher. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

process. WSMF recognizes the importance of mediation for the successful deployment of Web Services by making mediation a first class component. A mediator provides a link to a mechanism that can resolve the identified mismatch.

WSMF was instantiated in the form of an ontology for Web services, a family of formal languages and a reference architecture. WSMO (Fensel et al., 2006) provided a formal ontology for describing Web services based upon WSMF. In addition to the two principles outlined above, WSMO embodies a number of principles that are derived either from the UPML framework, and therefore PSMs research in general or from the principles underlying the Web and service-oriented computing.

4.1. Web and service-oriented computing-based principles

4.1.1. Web compliance

WSMO inherits the concept of URI for the unique identification of resources as the essential design principle of the World Wide Web. Moreover, WSMO adopts the concept of namespaces for denoting consistent information spaces (see http://en.wikipedia.org/wiki/namespace_(computer_science) for a definition), supports XML and other W3C Web technology recommendations, as well as the decentralization of resources.

4.1.2. Description versus implementation

Following the service-oriented computing principle that separates the interface of a Web service from its implementation WSMO differentiates between the descriptions of Web service elements (e.g., the types of the inputs) and the implementation of a Web service (in some programming language). Although the former requires a concise and sound description framework based on appropriate formalisms, the latter is concerned with the support of existing and emerging execution technologies for the Semantic Web and Web Services. WSMO aims at providing an appropriate ontological description model, and to be compliant with existing and emerging technologies.

4.2. PSM-based principles

4.2.1. Ontology based

After KADS, the use of separate domain models within PSM-based models became prevalent. Following from UPML, all resource descriptions as well as the descriptions of data interchanged during service usage are based on ontologies. The extensive usage of ontologies facilitates semantically enhanced information processing as well as providing support for interoperability.

4.2.2. Ontological role separation

Frameworks such as KADS and UPML separate descriptions of tasks and generic reasoning capabilities, distinguishing between a desired outcome or intention and mechanisms able to achieve this. Within WSMO, this idea has been extended to account for the two main roles that are present within Web service usage scenarios (see Fig. 3): the client and the service provider. Actors, which fulfill these roles, will each be embedded within their own context, and each of these actors, or more generally clients, exist in specific contexts, and the underlying epistemology of WSMO enables these contexts to be differentiated.

4.3. WSMO top-level elements

We will now outline the main top-level elements for WSMO indicating their relationship to the principles underlying WSMO and WSMF, and to PSMs. The complete WSMO specification can be found in Roman et al. (2006). The elements of the WSMO ontology are defined in a meta-meta-model language based on the Meta Object Facility (OMG, 2002).

4.3.1. Ontologies

Ontologies provide the formal semantics for the terminology used within all other WSMO components.

```
Class ontology
   hasNonFunctionalProperties type nonFunctionalProperties
   importsOntology type ontology
   usesMediator type ooMediator
   hasConcept type concept
   hasRelation type relation
   hasFunction type function
   hasInstance type instance
   hasAxiom type axiom
```

Following from the extensive use of metadata within the Web, every WSMO element includes a *nonfunctional properties* attribute that extends the Dublin Core Metadata Set. Nonfunctional properties include basic information such as the author and creation date and service-specific properties related to the quality of the described service.

Imported ontologies allow a modular approach for ontology design and can be used as long as no conflicts need to be resolved between the ontologies. When importing ontologies in realistic scenarios, some steps for aligning, merging, and transforming imported ontologies to resolve ontology mismatches are required. For this reason *ontology mediators* are used (OO Mediators). As mentioned earlier, the concept of mediators within WSMO is an extension of the UPML bridge concept.

The other elements are as normally found within ontology definition languages. *Concepts* constitute the basic elements of the agreed terminology for some problem domain. *Relations* are used to model interdependencies between several concepts (respectively, instances of these concepts); *functions* are special relations, with a unary range and an n-ary domain (parameters inherited from relation), where the range value is functionally dependent on the domain values, and *instances* are either defined explicitly or by a link to an instance store, that is, an external storage of instances and their values.

4.3.2. Web services

An online component that provides functionality is described using the service concept. The following listing presents the common elements of these descriptions.

```
Class service
   hasNonFunctionalProperties type nonFunctionalProperties
   importsOntology type ontology
   usesMediator type {ooMediator, wwMediator}
   hasCapability type capability multiplicity = single-valued
   hasInterface type interface
```

Within the service class the nonfunctional properties and imported ontologies attributes play a role that is similar to that found in the ontology class with the minor addition of a quality of service nonfunctional property. An extra type of mediator to deal with protocol and process related mismatches between Web services is also included.[1]

The final two attributes define the two core WSMO notions for semantically describing Web Services: a *capability* and *service interfaces.* A capability contains the following four main items:

1. *Preconditions*: This is a set of logical expressions that specify constraints over the inputs of a service. The focus here is on the data that can be accessed within the available reasoning system.
2. *Assumptions*: Within service usage scenarios one often needs to make statements about the world outside of the platform on which a service is executed. Although, of course, it is not always feasible to check the status value for the statements, the statements are still of value in terms of a formal description of a potentially important constraint. For example, in an online purchase scenario one may wish to make statements relating to the number of items in stock and the current balance of the customer.
3. *Postconditions*: This is a set of logical expressions that specify constraints over the outputs of a service. As above, the focus here is on the data that can be accessed within the available reasoning system.
4. *Effects*: These are statements that relate to the state of the world after the service has been executed. As with assumptions, it may not always be feasible to check the absolute truth values of the statements, but still they serve a useful formal documentation role and can

[1] One may consider that the use of mediators may cause efficiency problems. For our descriptions, we consider explicitness as a key factor.

facilitate verification and monitoring. For example, one may wish to make statements related to the physical location of a purchased item and the capacity of related warehouses and transportation resources.

Note that the refinement of the pre- and postconditions from earlier work on PSMs to the split between considerations of the state of the computer system and the potential status of the real world is mandated by the change in the domain of discourse. The concept of PSMs were generated to be used within a single reasoning engine that had access to all available knowledge. As mentioned earlier in service-oriented computing, two significant aspects of the environment are altered. First, the components are distributed over the Internet. Note that for the same Goal (or task) a different set of Web services may be invoked for nonfunctional reasons such as availability, cost, or quality of service. Second, Web services often act as proxies for business services, and are therefore embedded within an overall business environment. It is often important to capture the relevant facets of the business domain within the service description. However, we are hampered by the fact that here we are taking into account properties and values that we cannot access. Real-world considerations are also embodied within WSMO's nonfunctional properties.

A service interface consists of a *choreography* and an *orchestration.* A choreography can be seen as an extension of the PSM notion of input and output roles. Invoking a Web service can involve a complex set of interactions. For example, a client may need to login and send credit card details at the start of an interaction. In addition, clients may have interaction constraints. For example, only supplying credit data at the very end of an interaction. Most PSM-related frameworks did not consider interaction assuming that each PSM was a "single-shot" returning a single result. However, CommonKADS included a Communication Model where dialog diagrams were used to capture human–agent and interagent interaction.

An orchestration captures the control and data flow within a complex Web service that is somewhat similar to the control and data flow specifications commonly found in PSM-based descriptions. Within a service context orchestrations are commonly used to ensure behavioral congruence, that is, that the orchestration of a service matches its declared choreography; facilitate the reuse of service combinations; and enable client constraints to be checked. For example, a client may require that only ethical banking services are used in an interaction.

```
Class Goal
   hasNonFunctionalProperties type nonFunctionalProperties
   importsOntology type ontology
   usesMediator type {ooMediator, ggMediator}
   requestsCapability type capability multiplicity = single-valued
   requestsInterface type interface
```

4.3.3. Goals

WSMO goals are derived from the notion of task prevalent in previous work including KADS, UPML, and generic tasks (Chandrasekaran, 1986). Within a service-oriented computing context the use of goals is predicated on the fact that in many settings the underlying structure of service requests is reusable. For example, in vertical domains such as holiday booking requests will have a common structure (e.g., destination, length of stay, number of adults and children in the booking party).

Reflecting the Web service usage scenario shown in Figure 3, goals are used to represent the viewpoint of a service requester or client. Because a goal is the starting point for finding and composing relevant services, a WSMO goal also reflects the structure of a Web service capturing aspects related to user desires with respect to the requested functionality and behavior. Thus, the *requested capability* in the definition of a goal represents the functionality of the services the user would like to have, and the *requested interface* represents the interface of the service the user would like to have and interact with.

Goal reusability is supported by GG mediators that can be used to specify the difference between a reused goal and a desired goal specification. For example, a general Purchase Item Goal could be specialized to a Purchase Train Ticket Goal with associated constraints on the type of input and output.

4.3.4. Mediators

For every $1 spent on creating a software application, $5–$9 are spent on integration (http://www.ecommercetimes.com/story/20762.html?wlc=1220016853). Mediation in WSMO extends the notion of bridges in UPML, as mentioned earlier, and also builds upon the database notion of mediation as advocated by Wiederhold (1992). As with these two above approaches mediators in WSMO handles heterogeneities that can occur when two software components are put together. The core model shown below incorporates a number of sources, a target, and a mediation service that are capable of resolving the associated mismatch.

WSMO defines different types of mediators for connecting the distinct WSMO elements: *OO Mediators* connect and mediate heterogeneous ontologies, *GG Mediators* connect goals, *WG Mediators* link Web services to goals, and *WW*

Mediators connect interoperating Web services resolving mismatches between them.

The mediation service may be specified as a service, a WW mediator or as a goal. In the case of service or a WW mediator a relevant service is invoked. If a goal is specified then a discovery mechanism is invoked to find a relevant service.

```
Class mediator
   hasNonFunctionalProperties type nonFunctionalProperties
   importsOntology type ontology
   hasSource type {ontology, Goal, service, mediator}
   hasTarget type {ontology, Goal, service, mediator}
   hasMediationService type {Goal, service, wwMediator}
```

4.4. Comparing UPML and WSMO

A summary of the relationship between the UPML framework and WSMO is shown in Table 1.

4.5. Semantic Execution Environments (SEE)

Within a number of projects over the last few years we have been developing the concept of a SEE (Norton et al., 2008) based upon WSMO. A SEE can be viewed as a semantic broker for Web services following on from the concept of an intelligent broker for knowledge components as expounded in the IBROW3 project (Benjamins et al., 1998). The scenario within which SEEs are used can be seen in Figure 6, where four layers of IT infrastructure are involved as follows:

1. *Legacy system layer*: This layer is a set of heterogeneous IT systems distributed over the Internet involving a variety of organizations with no central control.
2. *Service abstraction layer*: This layer exposes the (micro-)functionality of the legacy systems as Web services, abstracting from the hardware and software platforms. This layer utilizes standard XML-based languages and protocols.
3. *SEE*: We expand on this layer below, but suffice it to say that a set of internal services uses the available WSMO descriptions to mediate between requests from the presentation layer and the available services.
4. *Presentation layer*: This layer is a Web application accessible through a standard Web browser that is built upon the semantic Web services layer. The WSMO goals defined within the semantic Web services layer are reflected in the structure of the interface and can be invoked through the SEE application program interface (API). We should emphasize that the presentation layer may comprise a set of Web applications to support different user communities. In this case each community would be represented by a set of goals supported by community related ontologies.

We have been working on producing a standard for SEEs within the OASIS SEE Technical Committee (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=semantic-ex) and an overview of the components can be seen in Figure 7.

Figure 7 reveals that a SEE consists of three main layers. The first is a *problem solving layer*, which consists of the following:

1. *ontologies*;
2. *applications*, for example, e-tourism and e-government; and
3. *developer tools*, which are graphical user interface tools such as those for engineering ontology/Web service descriptions, and generic developer tools such as language APIs, parsers/serializers, and converters.

The second layer is a *broker layer* consisting of the following:

4. *discovery*: finding services relevant for a given context;
5. *adaptation*: altering a service to improve its fit for a specific scenario;
6. *composition*: aggregating services using techniques such as planning;
7. *choreography*: handling communication between the broker and associated clients and services that are used;
8. *mediation*: incorporating (a) ontology mediation, which are techniques for combining ontologies and for overcoming differences between ontologies; and (b) process mediation for overcoming differences in message ordering;
9. *grounding*: mapping between the semantic, ontology-based descriptions and the XML formats used by the deployed services; and
10. *fault handling*: handling and recovering from faults within composed services incorporating multiple providers; and
11. *monitoring*: making the execution behavior visible and inspectable.

The third layer is a base layer that supports the exchange formalism used by the architecture, comprising the following:

12. *formal languages*: static ontology and behavioral (i.e., capability/choreography/orchestration) languages, for example, Web Service Modeling Language;
13. *reasoning techniques*: for reasoning over formal descriptions; LP, DL, and FOL behavioral languages; and

Table 1. *Summary of the relationship between the UPML framework and WSMO*

UPML Concept	WSMO Concept	Comment
	Nonfunctional properties	Derived from the Web concept of metadata
Domain model	Ontology	Ontologies in WSMO can contain generic concepts to represent the structure of a domain and be instantiated to capture specific data.
PSM	Web service	The main differences between a Web service and a PSM are that Web services are distributed, can represent business services, and invocation can involve complex interactions. Both include a notion of input and output and pre- and postconditions.
Task	Goal	A task specifies a knowledge objective to be achieved, whereas a goal reflects the viewpoint of a Web service client and supports the reuse of service requests. Both include a notion of input and output and preconditions.
Input and output roles	Choreography	Choreographies enable the specification of the constraints over the interaction patterns involved in service communication.
Bridge	Mediator	Bridges within UPML were used to link different types of knowledge models (e.g., a task model to a PSM), whereas mediators in WSMO resolve mismatches between different WSMO entities (e.g., a goal to a Web service).
Precondition	Precondition and assumption	WSMO differentiates between statements concerning the accessible state of the IT system and statements related to the state of the world.
Postcondition	Postcondition and effect	As above

14. *storage and communication*: low-level management of data and internal communication.

Finally, vertical services such as (15) *execution management* and (16) *security* authentication/authorization, encryption, and trust/certification.

5. FUTURE OF PSMS

We can see that previous work on PSMs has made a significant impact in the area of service-oriented computing in enabling scalability and interoperability. We also envisage that this work will lead to three sizeable future contributions.

5.1. Enterprise level computing

Within a number of scenarios we have found the need to connect disparate domains of discourse through the use of semantics within enterprise scenarios.

Managing business processes is an extremely important activity for enterprises, and the challenges associated with this activity are addressed within the field of Business Process Management (BPM), an intersection between the fields of management and computing. One of the key challenges in BPM is centered on the fact that the area comprises two distinct domains of knowledge. The expertise for defining and managing business processes resides in the head of business analysts, who for the most part, have a background and experience in business management rather than in Information and Communication Technology (ICT). In contrast, a growing percentage of business processes now reside in ICT systems, and obviously, the knowledge required to manage enterprise infrastructures resides in the head of computer specialists. A notable challenge thus within BPM is *bridging the business–ICT process gap.*

Within the large European Project SUPER (http://www.ip-super.org), we have been constructing a methodology, architecture, and a suite of tools based upon a Semantic Business Process Management (SBPM) approach (Hepp et al., 2005): a combination of BPM, Semantic Web, and Web Services technologies. SBPM is a multilayered approach where a number of standard languages and notations have been mapped to a stack of ontologies supported by a suite of semantically enhanced tools. By analyzing requirements from our industrial partners within SUPER we have derived the following main SBPM-related requirements:

1. *Supporting the design of business process models*: Adding formal semantics to business process models enables business analysts to
   a. find relevant existing process models for solving a business task, which match a given business context (e.g., business domain regulations or organizational policies);
   b. create new processes through the composition of processes exposed as Semantic Web Services; and
   c. mediate between incompatible processes that are required to be to be connected.
2. *Generating an executable process model*: Use ontological descriptions to move from an informal (usually diagram based) business-level process model to a model that can be executed within an engine.
3. *Monitoring and analyzing the progress of a running process*: Provide semantically rich information on the status of currently running processes, within a corporate ICT infrastructure, in a fashion that is understandable to the business analyst.

Within SUPER we use a five-layered approach to bridge the business-ICT process gap:

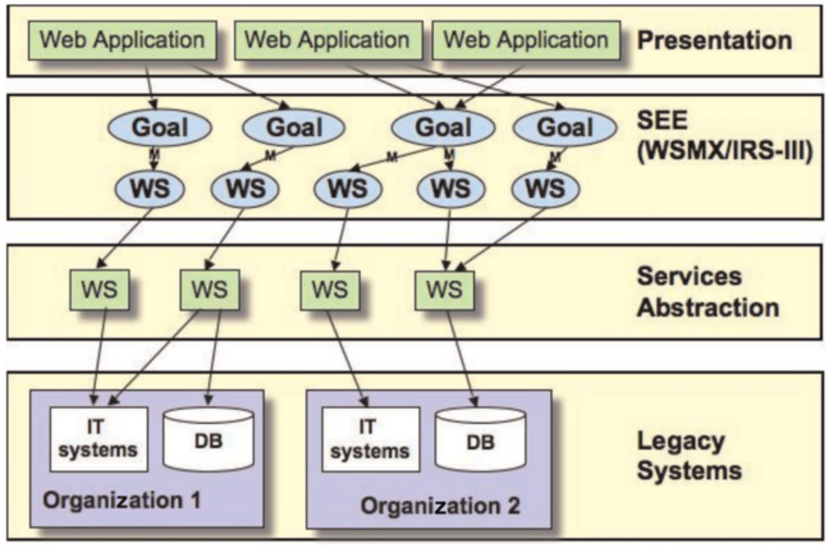

Fig. 6. The generic scenario for a semantic execution environment. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

1. *Business process strategic layer*: This is the highest level of abstraction of business processes, and represents business value chains and activities at the strategic enterprise level. Activities in this layer have traditionally relied on tools such as SAP's Solution Maps (http://www.sap.com/solutions/businessmaps/solutionmaps), which help business analysts to visualize and plan an integrated ICT solution. In SBPM we represent these activities through a set of organizational ontologies.
2. *Business process modeling layer*: This layer represents business process modeling activities. A number of current notations exist for capturing the knowledge of business analysts in diagrammatic form. Within super we focus on two: event-driven process chains (EPCs; Keller et al., 1992) and business process modeling notation (BPMN; OMG, 2006), which are typically embedded within tools. A number of ontologies have been developed to capture each specific notation (sEPC and sBPMN, respectively) in addition to a generic ontology to capture common abstractions: the Business Process Modeling Ontology (http://www.ip-super.org/ontologies/super/super_bpmo1_5.wsml).
3. *Business process workflow execution layer*: This layer represents executable business processes, which can be run using ICT workflow engines. We take Business Process Execution Language (BPEL) as the foundation

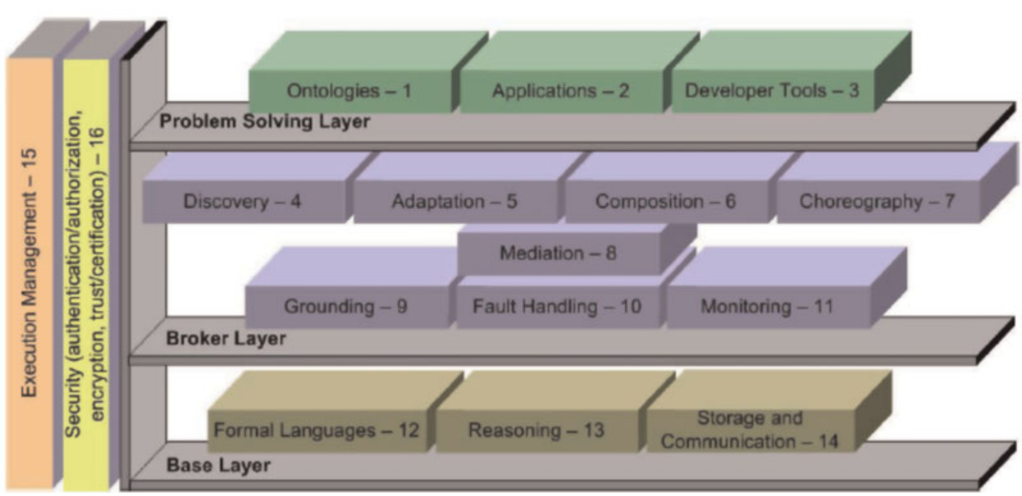

Fig. 7. The internal architecture of a semantic execution environment. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

for representing business processes in an executable workflow-based format and within SUPER an ontological reflection of BPEL (sBPEL) has been created (Nitzsche et al., 2007).

4. *Semantic Web services layer*: This layer uses a combination of WSMO and the SEE platform.

A set of transformations is currently being developed to map between each of the layers above. Top-down we semi-automatically transform business-oriented ontology-based descriptions first into a BPMO-based business processes and then into an executable workflow based on sBPEL. Bottom-up transform events within a BPEL execution are transformed into key performance indicators: enterprise-specific financial and nonfinancial metrics used to help an organization define and measure progress toward organizational goals (Medeiros et al., 2007). Again, our transformations are semantics based using a set of business process analysis ontologies (Pedrinaci et al., 2008).

Our current work is focused on using the heuristic classification PSM as a high-level framework for our transformations. In general, we see that there is a range of activities within enterprise IT settings that require two-way transformations equivalent to the abstraction and refinement processes defined within heuristic classification.

Within a second scenario we are extending the heuristic classification model to enable business applications (BAs) to be offered as utilities. A BA is a complete IT solution that encompasses a self-contained set of service descriptions, business processes, and additional artefacts (e.g., business rules and graphical user interfaces) that when combined represent a solution for a given complex business problem. In addition, a BA contains configuration data (such as deployment descriptors) and relevant nonfunctional properties (such as service level agreements and service level objectives).

In addition to the use of BAs the central notion within our work is that of a *template*. Templates simultaneously represent blueprints of business requirements (application templates) and descriptions of IT utility capacity (environment templates). Business solutions comprise a combination of application and environment templates instantiated for a particular context.

The heuristic classification-derived process model supporting BA construction from templates is depicted in Figure 8. The overall process model incorporates two points of abstraction. First, concrete business requirements, typically from an outsourcing client, are abstracted to a preexisting application template. Second, from the application hosting perspective specific constraints on the availability and use of hardware and software resources are abstracted to environment templates. A heuristic matching process finds pairs of application and environment templates that are compatible, given the context of the current request. The application and environment template pair forming the best fit are then coupled to form a business application model. Coupling binds the application and environment templates through a process that unifies common points of variability. In the next phase, a variety of refinement and instantiation techniques transform the business application model into a business solution. Finally, a provisioning process (http://en.wikipedia.org/wiki/provisioning) deploys a provisioned system from the descriptions contained within the derived business solution.

The refinement and instantiation techniques referred to above are based on configuration and parametric design PSMs. Configuration design (Darr et al., 1998) is a form of design where a predefined set of components are assembled such that input requirements are fulfilled and no constraints are violated. Configuration design differs from other forms of design in that no new components can be designed and the given set of requirements and constraints are assumed to be complete. In a number of cases, for example where no complex spatial requirements are involved, we are able to simplify our configuration design technique down to parametric design (Motta & Zdrahal, 1996). Within parametric design a set of target parameters are assigned values according to a set of requirements and constraints.

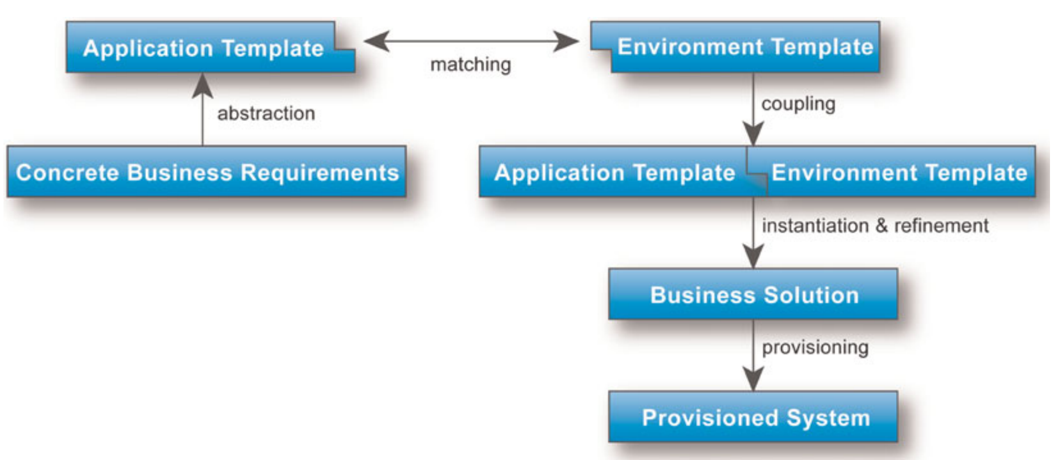

Fig. 8. A heuristic classification based life cycle for transforming business applications into utilities. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

We can see from the above that within enterprises there are scenarios that require knowledge transfer between the business level and the IT level. We have found that PSMs, such as heuristic classification, provide a solid knowledge level basis for structuring the framework for the overall mapping. In addition, PSM-derived ontologies, such as WSMO offer an encapsulation for a service-based IT level that is amenable to being mapped to a business level. Specifically, the knowledge-level split between desired goals or tasks and computational techniques or methods separates the concerns of business experts and IT specialists.

5.2. A web of services

Although very successful within enterprise settings, Web services, however, have failed to make a significant impact on the Web. Current estimates are that the Web currently contains 30 billion Web pages with 10 million new static pages added each day. In contrast, only 27,000 "true" Web services (http://seekda.com/, Web services with active end points described with a WSDL file) currently exist on the Web. Within the SOA4All project (http://www.soa4all.eu/) we aim to alleviate the above and realize a world where billions of parties are exposing and consuming billions of services via advanced Web technology. To achieve this the SOA4All framework combines four distinct paradigms:

1. service-oriented architectures as a means to abstract from software to service ware,
2. Semantic Web technology as a means to abstract from syntax to semantics to mechanise relevant tasks associated with Web service management and use,
3. Web technology as an infrastructure and underlying infrastructure for the integration of services at a worldwide scale, and
4. Web 2.0 as a means to structure human–machine cooperation in an efficient and cost-effective manner.

5.2.1. Web technology

We have already discussed the first two previously. We combine SOA and Semantic Web technology through the use of WSMO and our SEE infrastructure. From the Web we borrow the following principles:

- *Openness* implies that, in principle, anybody can contribute content as a provider or consumer of information. Openness is a major and essential necessity to ensure the success of our SOA4All platform. Usage of this infrastructure as a service provider or user is kept as simple, smooth, and unrestricted as possible.
- *Interoperability* is needed and provided through the integration of different proprietary and legacy solutions through a common interface. The solution as such must be platform and vendor neutral to enable all types of providers and requesters of information to participate.
- *Decentralized changeability* and dynamicity implies that content can appear, becoming modified, or disappear in an uncontrolled fashion. That is, the provisioning and modification of content should be through the distributed control of the peers rather by a central authority. Central control would hamper access and therefore scalability, an element of chaos or "messiness" must be tolerated.
- *Central means to route requests or responses must be automated* in order to scale. Manually generated repositories are inherently nonscalable, costly, and immediately become outdated. One could argue that Web sites like Google are actually a centralized control or access. However, what Google-type systems implement is an abstraction over access and a caching mechanism. In the early days of the Web sites were accessed via magic numbers (and later by magic names) and list of bookmarked pages were considered valuable intellectual property. Through search engines this access is replaced by key word retrieval and relevance-based ranking. Therefore, Google simply lifts the address bar of a browser from a URL to a higher level of abstraction: a keyword.
- *Enabling n:m relationship* to maximize interaction. In contrast to e-mail, where the content is targeted to specific receivers, the Web is based on anonymous distribution through publication. In principle, the information is disseminated to any potential reader, something mailing tries to achieve trough spam.

Instantiating the Web principles to fulfill the SOA4All goal requires *three important means*:

- A web of services requires a *worldwide addressing schema*. At an intermediate level it may be a unique name, and at a more elaborated level it may be a description of the capability of a service, that is, the degree to which it can be used to achieve a certain goal. In the case of the Web, URIs provide the addressing schema.
- A *transport layer* (a protocol) to transmit requests for and the results of services. In the case of the Web, this is HTTP.
- A platform independent *interface* to process service request and access. In case of the Web, this is HTML and browsers that interpret HTML.

5.2.2. Web 2.0

Although from a pure technological point of view the mechanisms used within Web 2.0 are similar to the "standard" Web, Web 2.0 brings a number of Web-related concerns to the fore that can facilitate the creation of a web of billions of services, including the following:

- blurring the distinction between content consumer and content provider,
- providing a lightweight semantic mechanism through tagging,

- blurring the distinction between service consumer and service provider, and
- blurring the distinction between machine- and human-based computation.

Web 2.0 technology can thus provide a means to easily generate and access the semantic service layer outlined above. Incorporating human interaction and cooperation in a comprehensive fashion creates a route to solving certain tasks such as service ranking or mediation that otherwise remain computationally infeasible. In a number of different scenarios Web 2.0 and human computing approaches together with their underlying social consensus building mechanisms have proven the potential of the appropriate combination of services offered by humans and services provided via automated reasoning. Thus, in the end, a service need not necessarily be supplied by a computer program and, for example, enable current approaches to service discovery and (human) expert finders to be combined.

The above demonstrates how the notion of PSMs applied within large-scale distributed heterogeneous contexts, through WSMO and SEE, provides a solid foundation for achieving a significant impact.

5.3. PSMs as the basis for a future Internet

With over a billion users worldwide, the current Internet is a great success: a global integrated communications infrastructure and service platform underpinning the World economy and society in general. However, today's Internet was designed in the 1970s for purposes that bear little resemblance to current and foreseen usage scenarios. Mismatches between original design goals and current utilisation are now beginning to hamper the Internet's potential. A large number of challenges in the realms of technology, business, society and governance have to be overcome if the future development of the Internet is to sustain the networked society of tomorrow.

A number of new initiatives are now underway to address these challenges. Within the United States the GENI project (http://www.geni.net/) adopts a clean slate approach to design and aims to develop a shared experimental facility for promoting research and development in the area of new Internet architectures and network services. GENI foresees a common network foundation enabling multiple network experiments to be conducted simultaneously and independently. It aims to resolve problems of the existing Internet architecture concerning stability, security, quality of service; to construct experimental environments on actual networks using new network technologies, and to incorporate innovative technologies such as optical, mobile, and sensor technologies.

In Japan the AKARI Architecture Design Project (http://akari-project.nict.go.jp/) aims to implement a new generation network by 2015, developing a network architecture and creating a network design based on that architecture. As with GENI, within AKARI new network architectures will be generated using a clean-slate approach. Once these new network architectures have been designed, the issue of migration from the current Internet will be considered using the generated design principles.

Within Europe a "Future Internet" (http://www.future-internet.eu/) initiative combines the work of 70 projects within the Framework 7 Programme and was kicked off with a conference in March 2008 in Bled, Slovenia (http://www.fi-bled.eu/). Following the conference, six working groups were set up that are collaborating along the dimensions of: experimental facilities, Internet of things, multimedia content, security and trust, and service-oriented architectures.

Abstracting from the various initiatives, we find two foundational layers are associated with the Future Internet: a network layer and a service layer.

5.3.1. Future Internet network layer

The network layer will incorporate hardware for routing and low-level protocols that support "standard" PCs as well as portable devices including mobile phones. Most work, therefore focuses, on innovations in areas such as photonics, wireless communication, and routing solutions. Proposals have been made to incorporate semantics into the network. Most notably, Clark et al. (2003) propose the creation of a knowledge plane (KP) incorporating a "cognitive framework" that is able to learn and reason. The KP would support the following:

1. *Fault diagnosis and mitigation*: Users are currently not able to find out why an Internet failure has occurred or how the fault may be corrected. The KP would support end-user explanation for errors and include an element of automatic fault diagnosis and repair.
2. *Automatic (re)configuration*: Routing on the Internet today is often defined by static policy tables through manual configuration, which do not take administrative or policy constraints into account. A KP configuration manager would use high-level assertions to control network regions or facilitate the dynamic creation of ad hoc networks. A key issue here would be the use of two-way mappings between high-level policy knowledge and low-level network settings to support control and explanation of network state.
3. *Support for overlay networks*: An overlay network is a virtual network built on top of an existing one. Overlay networks are useful in situations where one requires the provision of a specific set of functionalities. The KP would support high-level network performance information supporting individual application requirements. In this scenario each application would have its own specifically tuned network through the aggregation of application- and network-derived knowledge
4. *Knowledge-enhanced intrusion detection*: Detecting network intrusions is based upon the detection of specific data patterns within a network. The KP would enhance intrusion detection by integrating and correlating

heterogeneous data from several points in the network reducing the generation of false positives and false negatives.

We can see from the above that scope exists for the application of PSMs. Heuristic classification has successfully been applied in fault diagnosis and Propose and Revise (Marcus et al., 1988) has been used to create repair plans. Configuration and Parametric Design can support automatic (re)configurating, allowing domain knowledge related to administrative policy and network state to be taken into account.

As mentioned above, a key requirement for supporting overlay networks within the KP is the aggregation of network and application-related knowledge. Aggregation and correlation of heterogeneous knowledge is also required for knowledge-enhanced intrusion detection. Notions of abstraction and refinement as contained within heuristic classification are useful for both these functionalities. In addition, heuristic classification provides a suitable framework for matching disparate network data patterns against known types of network intrusion. More generally, we can see above how PSMs facilitate a mapping from the low-level network data to a user and organizational knowledge level view as required by the KP.

5.3.2. *Future Internet of services*

The common term for the service layer within the Future Internet capturing the underlying spirit of the approach is the "Internet of Services," a continuity that combines two main perspectives, namely, those of the service consumer and the service provider.

1. *Service consumers* look for "perfect interactivity," where "perfect" means
   a. *permanent:* interactivity at anytime;
   b. *transparent:* the service consumer need only concentrate on the benefits of the service he or she is using;
   c. *seamless:* the interaction is performed using "typical" devices appropriate to the current context; and
   d. *trustworthy:* secure, private, and safe.
2. *Service providers* require new approaches to management where control is decentralized and the focus is placed on maintaining the consistency of each service.

WSMO-based service descriptions can support transparency, as service consumers need only specify high-level goals. Within the SOA4All project we are extending WSMO to support service contextualization in the following manner. A complex service can be described through an orchestration (a control and data flow specification) of goals. A service is instantiated for a specific consumer context through the matching of the orchestrated goals to appropriate Web services.

Trust is a key grand challenge as a whole for the Future Internet, and we describe how PSMs can support this below.

In Section 4.5 we described how SEEs, based on WSMO, can support the management of services, semiautomating tasks such as service discovery and invocation. One of the main principles underlying WSMO (see the beginning of Section 4) is the strict decoupling of components including their semantic descriptions.

In addition to the above new emerging trends including "software as a service" (SaaS) and "resource as a service" (RaaS) will feature in a Future Internet. In Section 5.1 we outlined how PSM-derived frameworks can link SaaS and RaaS to enterprise level viewpoints.

5.3.3. *Key future of Internet challenges*

Ongoing discussions within the Future Internet Working groups have formulated a number of challenges that need to be addressed if a successful Future Internet is to emerge.

1. *Scalability*: The increasing scale of the Internet brings new challenges in a number of areas. Examples of these challenges include modeling, validation, and the verification of business processes built on top of SOA; the flexible evolution and execution of business processes; data, process, and service mediation; the reliable management of composed services; and brokering, aggregation, and data management. Quality of software is an important factor in all of the above, and will become essential to the smooth operation of the "service universe." Approaches to tackling scalability include *openness:* lowering the barriers to entry so that large numbers are able to participate; and enabling the mechanization of certain tasks currently carried out by IT developers through semantic service descriptions such as WSMO.
2. *Trust*: Creating trusted environments for the new service world will require mechanisms to monitor, display, and analyze information flows between nodes participating in complex collaborations to detect and assess security risks and mechanisms to ensure trust and confidence in services created by end users themselves, that is, built-in safeguards and guarantees to maximize the trustworthiness of the new services. In addition, it is necessary to bring about changes in perception. P2P services today are too often associated with activities of doubtful legality, such as the illegal trading of copyright-protected content. Technical and legal mechanisms that encourage law-abiding attitudes need to be developed. In Galizia (2006) and Galizia et al. (2007) we outline a trust framework based on an extension to WSMO. Within the framework, Web Services Trust Ontology, heuristic classification is used to classify Web services according to their trustworthiness for a particular user and task. The classification is then used to aid service selection.
3. *Interoperability*: This applies at many different levels: service interoperability to provide an automated capability to integrate stand-alone services with services

that are similar or complementary, for instance, from a related business domain; data interoperability to provide the automated understanding of the information exchanged and ensure the overall quality of the service; and interoperability of the service layer with the network and application layers of different providers. In addition, semantic interoperability is important from a quality of service perspective to facilitate composition and middleware support. Our approach to tackling this challenge is based on WSMO mediators (see Section 4.3), which are derived from the bridges found within the UPML framework. WSMO mediators allow one to specify locations requiring data, conceptual, protocol, or process interoperability.

4. *Pervasive usability*: Services will be exploited by end users through different devices in different contexts of use. This will require the development of flexible mechanisms for adapting the user interaction to the current context of use while providing consistent user interfaces. Moreover, it will require the maintenance of state within user sessions even when a change of device is involved. This feature will support users in seamlessly carrying out their tasks through different devices in a manner more natural than possible presently. As mentioned above, our approach within the SOA4All project to supporting service contextualization is based upon the instantiation of orchestrations of goals to suit specific user scenarios. More generally, WSMO representations of services coupled with semantic user profiles and state.
5. *Mobility*: Mobile users will require instant, on-the-fly service creation providing a stable front end to a volatile dynamic aggregation of networks and services. Issues involved here include personal context, efficient utilization and discovery, ease of accessibility and mobile-to-mobile communication. As mentioned earlier WSMO-based descriptions can support the automation of tasks associated with service management and use. The principle of ontologically decoupling the viewpoints of service consumes and providers enhances the ability to create stable front ends, based on goals, while at the back-end services appear and disappear.

Added to the above, implementing the Future Internet will also require alignment between the viewpoints, actions, and strategies amongst the Telco, Media, and IT domains to support the needs of business and users. Our view is that PSM derived technologies can aid in bridging between these disparate areas.

6. CONCLUSIONS

Historically, PSMs provided a solid foundation for investigating how knowledge systems could be created by reusing domain-independent heuristic reasoning components. Libraries of PSMs have been created based upon generic PSM-based frameworks such as UPML. Genericity and reusability have been demonstrated through the diverse set of applications created ranging from medical diagnosis, room allocation (Shadbolt et al., 1993) and lift configuration (Motta et al., 1996).

We now live in a networked age where the only viable approach to managing the sheer scale of data created and consumed by the 1.4 billion Internet users is through a combination of Web service and semantic technologies. Recent work has seen a derivation of the UPML framework created to semantically describe services in the Web context. The epistemology of WSMO partitions the domain of services into Goals, to describe a Web-service client, Mediators, to manage mismatches, and Web Services to capture provided functionality and behavior.

The SEE is an abstract architecture, currently being standardized within OASIS, for brokering between service invocations and deployed services described using WSMO. Applications developed using SEE implementations (Cimpian et al., 2005; Domingue et al., 2008) have been created within real-world use within a number of European projects, such as DIP (http://dip.semanticweb.org/), demonstrating the feasibility of the approach (see, e.g., Tanasescu et al., 2007; Gugliotta et al., 2008). It is only through semantically enhanced infrastructures based on architectures such as SEE that Web services will begin to fulfill their initial promise.

Building on the semantically enhanced service frameworks and infrastructures described above we see a promising future for the continuation of PSM-derived work along three main themes. Managing large numbers of enterprise processes derived from business level goals and constraints and implemented in IT systems will benefit from WSMO-based research. WSMO provides a firm foundation with which to describe processes at the IT level that can then be mapped to ontological reflections of business oriented notations. The advent of cloud computing (http://en.wikipedia.org/wiki/Cloud_computing) will see the use of process centric descriptions for automating the provisioning of business applications from enterprise level descriptions.

In the mid- and long term we see a combination of PSM-based semantics with technologies embodying the principles found in SOA, Web, and Web 2.0, facilitating the emergence of a Service Web where billions of parties use, manage, and exchange billions of services in a dynamic unconstrained fashion. We also see that the main challenges faced within the Future Internet (scalability, trust, interoperability, pervasive usability, and mobility) can only be tackled systematically through semantic-based frameworks such as those described above.

We live in a networked age, where data and users need to be managed at planetary scale, using technologies that hide complexity and enable users to specify what they want to achieve rather than how. In our view, PSMs provide a firm foundation for creating mechanisms able to meet the challenges associated with planet-scale systems and will usher in an age where computing technologies truly meets the needs of its users.

ACKNOWLEDGMENTS

We acknowledge the DIP, SUPER, SOA4All, and Service Web 3.0 European projects in which this work was carried out. This work has also greatly benefited from discussions we had within the WSMO and Services Future Internet Working Groups and with our colleagues within the Semantic Web Services group in the Knowledge Media Institute at Open University and at Semantic Technology Institute Innsbruck and Semantic Technology Institute International.

REFERENCES

Benjamins, V.R., Plaza, E., Motta, E., Fensel, D., Studer, R., Wielinga, B., Schreiber, G., Zdrahal, Z., & Decker, S. (1998). IBROW3: an intelligent brokering service for knowledge-component reuse on the world-wide web. *Proc. 11th Banff Knowledge Acquisition for Knowledge Based System Workshop (KAW98)*, Banff, Canada, 1998.

Chandrasekaran, B. (1986). Generic tasks in knowledge based reasoning: high-level building blocks for expert systems design. *IEEE Expert 1(3)*, 23–30.

Cimpian, E., Vitvar, T., Zaremba, M., Moran, M., & Oren, E. (2005, February). *Overview and scope of WSMX.* WSMX Working Draft D13.0v0.2.

Clancey, W.J. (1983). The epistemology of a rule based expert system—a framework for explanation. *Artificial Intelligence 20*, 215–251.

Clancey, W.J. (1985). Heuristic classification. *Artificial Intelligence 27*, 289–350.

Clark, D.D., Partridge, C., Ramming, C.J., & Wroclawski, J.T. (2003). A knowledge plane for the Internet. *Proc. ACM SIGCOMM*.

Crubézy, M., Musen, M.A., Motta, E., & Lu, W. (2003). Configuring online problem-solving resources with the Internet reasoning service. *IEEE Intelligent Systems 18(2)*, 34–42.

Darr, T., Klein, M., & McGuiness, D.L. (1998). Special Issue: configuration design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 12*, 293–294.

de Kleer, J., & Brown, J.S. (1984). A qualitative physics based on confluences. *Artificial Intelligence 24*, 7–83.

Domingue, J., Cabral, L., Galizia, S., Tanasescu, V., Gugliotta, A., Norton, B., & Pedrinaci, C. (2008). IRS-III: a broker based approach to Semantic Web services. *Journal of Web Semantics 6(2)*, 109–132.

Fensel, D., Bussler, C., Ding, Y., & Omelayenko, B. (2002). The Web services modeling framework WSMF. *Electronic Commerce Research and Applications 1(2)*.

Fensel, D., Lausen, H., Polleres, A., de Bruijn, J., Stollberg, M., Roman, D., & Domingue, J. (2006). *Enabling Semantic Web Services: The Web Service Modelling Ontology*. Berlin: Springer.

Fensel, D., & Motta, E. (2001). Structured development of problem solving methods. *IEEE Transactions on Knowledge and Data Engineering 13(6)*, 913–932.

Galizia, S. (2006). WSTO: a classification based ontology for managing trust in Semantic Web Services. *Proc. 3rd European Semantic Web Conf.*, Budva, Montenegro.

Galizia, S., Gugliotta, A., & Domingue, J. (2007). A trust based methodology for web service selection. *Proc. 1st IEEE Int. Conf. Semantic Computing*, Irvine, CA.

Gugliotta, A., Domingue, J., Cabral, L., Tanasescu, V., Galizia, S., Davies, R., Gutierrez Villarias, L., Rowlatt, M., Richardson, M., & Stincic, S. (2008). Deploying Semantic Web services based applications in the e-government domain. *Journal on Data Semantics 10*, 96–132.

Hepp, M., Leymann, F., Domingue, J., Wahler, A., & Fensel, D. (2005). Semantic business process management: a vision towards using Semantic Web services for business process management. *Proc. IEEE Int. Conf. e-Business Engineering (ICEBE 2005)*, pp. 535–540, Beijing.

Keller, G., Nuttgens, M., & Scheer, A.W. (1992). *Semantische Process-modellierung auf der Grundlage Ereignisgesteuerter Processketten (EPK)*. University of Saarland, Saarbrucken, Veroffentlichungen des Instituts fur Wirtschaftsinformatik, Heft 89.

McDermott, J. (1980). R1: an expert in the computer systems domain. *Proc. AAAI 1980*, pp. 269–271.

Marcus, S., Stout, J., & McDermott, J. (1988). VT: an expert elevator designer that uses knowledge based backtracking. *AI Magazine 9(1)*, 95–112.

McDermott, J. (1988). Preliminary steps toward a taxonomy of problem-solving methods. In *Automating Knowledge Acquisition for Experts Systems* (Marcus, S., Ed.). Boston: Kluwer Academic.

Medeiros, A., Pedrinaci, C., Aalst, W., Domingue, J., Song, M., Rozinat, A., Norton, B., & Cabral, L. (2007). An outlook on semantic business process mining and monitoring. *Proc. 3rd Int. IFIP Workshop on Semantic Web and Web Semantics (SWWS '07)*, Move Federated Conferences and Workshops.

Motta, E. (1999). *Reusable Components for Knowledge Modelling*. Amsterdam: IOS Press.

Motta, E., Domingue, J., Cabral, L., & Gaspari, M. (2003). IRS-II: a framework and infrastructure for Semantic Web services. *Proc. 2nd Int. Semantic Web Conf. (ISWC 2003)*. Lecture Notes in Computer Science, Vol. 2870. Berlin: Springer.

Motta, E., O'Hara, K., Shadbolt, N., Stutt, A., & Zdrahal, Z. (1996). Solving VT in VITAL: a study in model construction and knowledge reuse. *International Journal of Human–Computer Studies 44*, 333–371.

Motta, E., & Zdrahal, Z. (1996). Parametric design problem solving. *Proc. 10th Knowledge Acquisition for Knowledge Based Systems Workshop*, Banff, Canada.

Musen, M.A., Fergerson, R.W., Grosso, W.E., Noy, N.F., Crubézy, M., & Gennari, J.H. (2000). Component based support for building knowledge–acquisition systems. *Proc. Conf. Intelligent Information Processing (IIP 2000) of the Int. Federation for Information Processing World Computer Congress (WCC 2000)*, pp. 18–22. Accessed at www-smi.stanford.edu/pubs/SMI_Reports/SMI-2000-0838.pdf

Newell, A. (1982). The knowledge level. *Artificial Intelligence 18*, 87–127.

Nitzsche, J., Wutke, D., & van Lessen, T. (2007). An ontology for executable business processes. *Proc. Workshop on Semantic Business Process and Product Lifecycle Management (SBPM-2007)*, CEUR-WS.

Norton, B., Pedrinaci, C., Domingue, J., & Zaremba, M. (2008). Semantic execution environments for semantics-enabled SOA. In *IT-Methods and Applications of Informatics and Information Technology*, Vol. 2, pp. 118–121. Oldenbourg, Germany: Wissenschaftsverlag.

OMG. (2002). *The object management group: meta-object facility, version 1.4.* Accessed at http://www.omg.org/technology/documents/formal/mof.htm

OMG. (2006). *Business process modeling notation specification 1.0.* Accessed at http://www.bpmn.org/Documents/OMG%20Final%20Adopted%20BPMN%201-0%20Spec%2006-02-01.pdf

Pedrinaci, C., Domingue, J., Brelage, C., van Lessen, T., Karastoyanova, D., & Leymann, F. (2008). Semantic business process management: scaling up the management of business processes. *Proc. Int. Conf. Semantic Computing (ICSC 2008)*, Santa Clara, CA.

Pedrinaci, C., Domingue, J., & Medeiros, A. (2008). A core ontology for business process analysis. *Proc. 5th European Semantic Web Conf.*, Tenerife, Spain.

Roman, D., Lausen, H., Keller, U., de Bruijn, J., Bussler, C., Domingue, J., Fensel, D., Hepp, M., Kifer, M., König-Ries, B., Kopecky, J., Lara, R., Oren, E., Polleres, A., Scicluna, J., & Stollberg, M. (2006). *Web Services Modeling Ontology (WSMO).* WSMO Working Draft D2v1.3. Accessed at http://www.wsmo.org/TR/d2/v1.3/

Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W., & Wielinga, B. (1999). *Knowledge Engineering and Management: The CommonKADS Methodology*. Cambridge, MA: MIT Press.

Shadbolt, N., Motta, E., & Rouge, A. (1993). Constructing knowledge based systems. *IEEE Software 10(6)*, 34–38.

Shortliffe, E.H. (1974). *MYCIN: a rule based computer program for advising physicians regarding antimicrobial therapy selection.* PhD dissertation, Stanford University.

SOAP. (2003). *SOAP Version 1.2 Part 0: Primer.* Accessed at http://www.w3.org/TR/soap12-part0/

Tanasescu, V., Gugliotta, A., Domingue, J., Gutiérrez Villarías, L., Davies, R., Rowlatt, M., Richardson, M., & Stincic, S. (2007). Geospatial data integration with Semantic Web services: the emerges approach. In *The Geospatial Web* (Scharl, A., & Tochtermann, K., Eds.). New York: Springer.

UDDI. (2003). *UDDI spec technical committee specification v. 3.0.* Accessed at http://uddi.org/pubs/uddi-v3.0.1-20031014.htm

Wiederhold, G. (1992). Mediators in the architecture of future information systems. *IEEE Computer 25(3)*, 38–49.

Wielinga, B.J., Schreiber, A.Th., & Breuker, J.A. (1992). KADS: a modelling approach to knowledge engineering. *Knowledge Acquisition 4(1)*, 5–53.

WSDL. (2001). *Web Services Description Language (WSDL) 1.1.* Accessed at http://www.w3.org/TR/2001/NOTE-wsdl-20010315

John Domingue is Deputy Director of the Knowledge Media Institute at The Open University and President of Semantic Technology Institute International. He has published over 150 refereed articles in the areas of artificial intelligence and the Web. Dr. Domingue is Chair of the Steering Committee for the European Semantic Web Conference Series, Chair of the Future Internet Symposium Series, and Co-Chair of the OASIS Technical Committee on Semantic Execution Environments. He also serves on the editorial boards for *International Journal of Human–Computer Interaction*, *Journal of Web Semantics*, and *Applied Ontology Journal*.

Dieter Fensel is a Professor at the University of Innsbruck and Director of the Semantic Technology Institute Innsbruck, a research institute. He also cofounded the Semantic Technology Institute International, which comprises leading research and industrial bodies who share a common vision of the future: a semantics-enabled Internet. Dr. Fensel has over 240 publications in the form of scientific books and journals, conferences, and workshop contributions. He co-organized more than 200 conferences and workshops and is a recipient of the Carl–Adam–Petri Award of the University of Karlsruhe. His current research interests focus on the development and application of semantics to all areas of computer science.