# Applying Semantic Web Services

## Conference or Workshop Item

# Applying Semantic Web Services

Stefania Galizia, Alessio Gugliotta, Carlos Pedrinaci, John Domingue
Knowledge Media Institute, The Open University,
Walton Hall, Milton Keynes, MK7 6AA
United Kingdom
{S.Galizia, A.Gugliotta, C.Pedrinaci, J.B.Domingue}@open.ac.uk

**Abstract.** The use of Semantic Web Services (SWS) for increasing agility and adaptability in process execution is currently investigated in many settings. The common underlying idea is the dynamic selection, composition and mediation - on the basis of available SWS descriptions – of the most adequate Web resource (services and data) to accomplish a specific process activity. In this paper we describe IRS-III, a framework for creating and executing semantic Web services, which takes a semantic broker based approach to mediating between service requesters and service providers. We describe the overall approach of IRS-III from an ontological perspective. We then illustrate our approach through three different applications to domains of Business Process Management, e-Learning and e-Science.

**Keywords:** Semantic Web Services, SWS Applications, Ontologies.

## 1 Introduction

The continuous diffusion of Web services increases the sharing of resources – services and data – on the Web. The specific nature of Web services - self-contained and platform-independent computational elements – gives them high availability and facilitates their reusability and interoperability across several application domains. One of the advantages of Web service technology is indeed the fairly simple aggregation of complex services out of repositories of simpler or even atomic services. However, Web service standards – [10], [12] and [14] - do not completely describe the capability of a service and cannot be understood by software programs. A human developer is thus required to interpret the meaning of inputs, outputs and applicable constraints, as well as the context in which services can be used. Therefore, the automatic discovery and selection at runtime of the most adequate resource for a given activity is limited, as well as the automatic solution of possible mismatches at the level of data format, message protocol and underlying organizational processes.

*Semantic Web Services* (SWS) research aims to automate the development of Web service based applications through the semantic Web technologies. By providing formal descriptions with well defined semantics, SWS facilitate the machine interpretation of Web service – functional and not functional - properties. The research agenda for SWS identifies a number of key areas of concern, namely:

- *Discovery:* finding the Web service which can fulfil a task. Discovery usually involves matching a formal task description against semantic descriptions of Web services.
- *Mediation*: we can not assume that the software components which we find are compatible. Mediation aims to overcome all incompatibilities involved. Typically this means mismatches at the level of data format, message protocol and underlying business processes.
- *Composition*: often no single service will be available to satisfy a request. In this case we need to be able to create a new service by composing existing components. Artificial Intelligence (AI) planning engines are typically used to compose Web service descriptions from high-level goals.

Significant results are already available, in terms of reference ontologies, e.g. OWL-S [9] and WSMO [5], comprehensive frameworks (e.g. DIP project[1] results), and more recently standards, e.g., SAWSDL[2]. Therefore, further research efforts are now investigating how SWS can be effectively applied – and in case improved - to solve other (Web-) service oriented computing problems.

In this paper we describe IRS-III (Internet Reasoning Service), a framework for creating and executing semantic Web services, which takes a semantic broker based approach to mediating between service requesters and service providers [2], [3]. More specifically, we have extended the core epistemological framework of our previous IRS-II framework [8] and incorporated the Web Services Modelling Ontology [5] conceptual model into the IRS-III framework.

In Section 2 we describe IRS-III specifically from an ontological point of view. In Section 3 we outline how SWS based systems can be successfully developed and deployed using IRS-III and we illustrate our approach through three different application domains: Business Process Management, e-Learning and e-Science. Section 4 concludes the paper.

## 2 IRS-III: A broker-based approach for SWS

IRS-III [2], [3] is a platform and broker for developing and executing SWS. A core design principle for IRS-III is to support capability-based invocation. A client sends a request which captures a desired outcome or goal and, using a set of semantic Web service descriptions, IRS-III will: a) discover potentially relevant Web services; b) select the set of Web services which best fit the incoming request; c) mediate any mismatches at the conceptual level; and d) invoke the selected Web services whilst adhering to any data, control flow and Web service invocation constraints.

To achieve this, IRS-III adopts a semantic Web based approach and is thus founded on ontological descriptions. At the heart of IRS-III there is the SWS Library, where semantic descriptions of various aspects of Web services, reference Domain Ontologies and Knowledge bases (instances) are stored using OCML representation

[1] http://dip.semanticweb.org/
[2] http://www.w3.org/2002/ws/sawsdl/

language [7]. Specific IRS-III components interpret such descriptions to discover and select the appropriate Web service, choreograph and ground to the Web service operations [4], orchestrate multiple Web services, and mediate semantic descriptions by running mediation rules or invoking mediation services [1]. Note that IRS-III supports grounding to standard Web services with a WSDL description, as well as stand-alone Java and Lisp code. Similarly, Web applications accessible as HTTP GET requests are handled internally by IRS-III.

### 2.1 IRS-III Service Ontology

The IRS-III service ontology forms the epistemological basis for IRS-III and provides semantic links between the knowledge level components describing SWS and the conditions related to its use. These descriptions are interpreted by the OCML reasoner. We describe the commonalities and differences between the service ontology and WSMO and then how the service ontology is used within IRS-III.

The IRS-III service ontology contains the following main items, which are also part of the Web Services Modelling Ontology [5]:

- Non-functional properties – these properties are associated with every main component model and can range from information about the provider such as the organisation's legal address, to information about the service such as category, cost and quality of service, to execution requirements such as scalability, security or robustness.
- Goal-related information – a goal represents the user perspective of the required functional capabilities. It includes a description of the requested Web service capability.
- Web service functional capabilities – represent the provider perspective of what the service does in terms of inputs, output, pre-conditions and post-conditions. Pre-conditions and post-conditions are expressed by logical expressions that constrain the state or the type of inputs and outputs.
- Choreography – specifies how to communicate with a Web service.
- Grounding – associated with the Web service choreography, a grounding describes how the semantic declarations are associated with a syntactic specification such as WSDL.
- Orchestration – the orchestration of a Web service specifies the decomposition of its capability in terms of the functionality of other Web services.
- Mediators – a mediator specifies which top elements are connected and which type of mismatches can be resolved between them.

The differences between our ontology and WSMO are described below:

- Meta-classes for the top-level SWS concepts – meta-class definitions for goal, mediator and Web service have been defined. These classes have a 'meta-' extension (e.g. meta-goal) and enable the IRS-III components to reason over the top-level concepts within the service ontology as first class entities.
- SWS user definitions as classes – following from the previous item, we enable users to define the required goals, mediators and Web services as subclasses of the corresponding WSMO concepts rather than as instances. In our view a class better captures the concept of a reusable service description and taxonomic

structures can be used to capture the constitution of a particular domain. For example, goals for booking flights may have sub-goals for booking European flights and for booking long-haul flights. A proposal for extending WSMO with goal templates, similar to our goal classes, has been suggested recently [11].

- SWS invocation contexts as instances – we reserve instances for invocation. When IRS-III receives a client request, instances of relevant goals, mediators and Web services are created to capture the current invocation context.
- Explicit input and output role declaration – in the interests of simplifying the definition of goals and Web services, our ontology incorporates explicit input and output role declarations. The declared input and output types are imported from domain ontologies. This feature enables SWS developers to view goals and Web services as 'one-shot' thus minimizing the need to consider complex interaction when appropriate.
- Orchestration and choreography language – the representation of our orchestration and choreography are defined within the service ontology.

Using SWS descriptions for implementing internal components, we implement several IRS-III internal components using the service ontology and OCML. Our assumption is that IRS-III components described through goals, mediators, and Web services and through ontological concepts and relations are easier to understand and maintain than if they were implemented purely in a programming language.

### 2.2 Using the Service Ontology

Before we describe the IRS-III server and its components we first highlight the main ways in which the service ontology is used to implement the core functionalities.

- Web services are linked to goals via mediators - if a wg-mediator associated with a Web service has a goal as a source, then this Web service is considered to solve that goal. An assumption expression can be introduced for further refining the applicability of the Web service.
- GG-mediators provide data-flow between sub-goals – in IRS-III, gg-mediators are used to link sub-goals within an orchestration and so they also provide dataflow between the sub-goals.
- Web services can inherit from goals - Web services which are linked to goals 'inherit' the goal's input and output roles. This means that input role declarations within a Web service are not mandatory and can be used to either add extra input roles or to change an input role type.

Client choreography – the provider of a Web service must describe the choreography from the viewpoint of the client. Within WSMO the choreography expresses a number of constraints which should not be violated when a deployed service is invoked. Within the IRS-III we evaluate the client choreography in order to interact with the deployed Web service.

Mediation services are goals – a mediator declares a goal as the mediation service which can simply be invoked. The required data transformation is performed by the associated Web service.

## 3 Creating Semantic Web Service Based Applications

Our generic application architecture is depicted in **Fig. 1**. As can be seen, the architecture is composed of four layers and enables collaboration between one or more stakeholders in a distributed fashion.

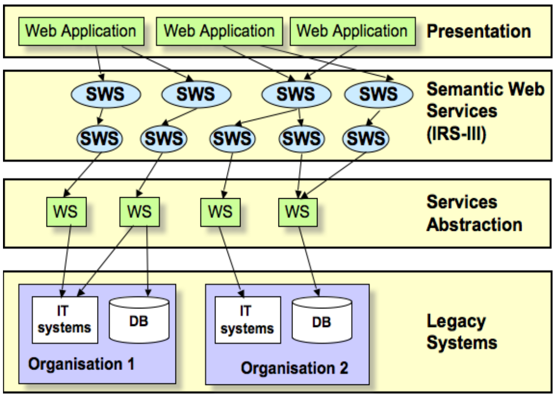


**Fig. 1**. The generic architecture used when creating IRS-III based applications

In particular, our approach enables the functionality provided by existing legacy systems from the involved business partners to be exposed as Web services, which are then semantically annotated and published using the IRS-III infrastructure. From the bottom up the four application layers are:

- Legacy system layer - consists of the existing data sources and IT systems available from each of the parties involved in the integrated application.
- Service abstraction layer - exposes the (micro-)functionality of the legacy systems as Web services, abstracting from the hardware and software platforms. In general existing Enterprise Application Integration (EAI) software will facilitate the creation of the required Web services. Note that for standard databases the necessary functionality of the Web services can simply be implemented as SQL query functions.
- Semantic Web services layer – given a goal request, this layer, implemented in IRS-III, will: a) discover a candidate set of services; b) select the most appropriate; c) resolve any mismatches at the data, ontological or process level; and d) invoke the relevant set of Web services satisfying any data, control flow and invocation requirements.
- Presentation layer – a Web application accessible through a standard Web browser which is built upon the semantic Web services layer. The goals defined within the semantic Web services layer are reflected in the structure of the interface and can be invoked either through the IRS-III API or as an HTTP GET request. We should emphasize that the presentation layer may be comprised of 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.

In order to successfully create applications from semantic Web services as depicted in **Fig. 1** above four key activities need to be carried out as follows:

1. Requirements capture – during this step the requirements for the overall application are captured. Although there is no prescribe methodology, the resulting documents should describe the stakeholders, the main users and roles, any potential providers for Web services, and any requirements on the deployed infrastructure and interfaces.
2. Goal description – using the requirements documents above, relevant goals are identified and described in IRS-III. During this process any required supporting domain ontologies will be created.
3. Web service description – descriptions of relevant Web services are created within IRS-III. Again, any domain ontologies required to support the Web service descriptions are defined.
4. Mediator description – mismatches between the ontologies used, and mismatches within and between the formal goal and Web service descriptions are identified and appropriate mediators created.

All of the above steps are carried out by the SWS application developer. The first two steps are user/client centric and therefore involve discussions with the relevant client stakeholders, whereas Step 3 will require dialogue with the Web service providers. Steps 2 and 3 are mostly independent and in the future we expect libraries of goals and Web services to become generally available to support re-use.

### 3.1 Business Process Management

Business Process Management (BPM) intends to support "business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information" [13]. BPM aims to support the complete life-cycle of business processes, however, by doing so BPM has made more evident the existing difficulties for obtaining automated solutions from high-level business models, and for analyzing the execution of processes from both a technical and a business perspective. The fundamental problem is that moving between the Business Level and the IT Level is hardly automated.

Semantic Business Process Management (SBPM) that is, the combination of Semantic Web and Semantic Web Services technologies with BPM, has been proposed as a solution for overcoming these problems [6]. SBPM aims at accessing the process space of an enterprise at the knowledge level so as to support reasoning about business processes, process composition, process execution, etc. Major efforts are currently devoted to pursuing the SBPM vision in the context of the European project SUPER[3]. The project follows a multi-layered 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.

Within this project IRS-III is playing several key roles centered on its capabilities as a Semantic Web Services platform and its strong ontology reasoning support. The

[3] http://kmi.open.ac.uk/projects/super/

research in IRS-III carried in the context of SBPM is focused in several issues such as:

- Supporting the design of business process models – Adding formal semantics to business process models enabling business analysts to:
    - Find relevant existing process models for solving a business task, which match a given business context (e.g. business domain regulations or organizational policies),
    - Create new processes through the composition of processes exposed as Semantic Web Services,
    - Mediate between incompatible processes which are required to be to be connected.
- Generating an executable process model – Using ontological descriptions to move from an informal (usually diagram-based) business-level process model to a model which can be executed within an engine.
- Monitoring the progress of a running process – providing semantically rich information on the status of currently running processes, within a corporate ICT infrastructure, in a fashion which is understandable to the business analyst.

### 3.2 E-Learning

E-Learning aims to support students to achieve a predefined learning outcome. Current approaches consider the usage of software systems – e.g. Learning Content Management Systems (LCMS) – that provide the learner with composite learning contents: the so called Learning Objects (LOs). Based on either proprietary or standard metadata, a LO usually defines the learning process - i.e. the sequence of activities the learner has to follow to achieve his/her learning objective – as well as the set of learning resources – data or services - associated to each activity of the process. Since metadata standards mainly rely on syntactic descriptions, human developers are needed to understand the intended meaning of the metadata and carry out manually the activities related to learning process composition and resource allocation. Therefore, current approaches limit the reusability of existing learning resources available on the Web and restrict the ability of a learning application to adapt automatically to specific learning requirements and learning contexts.

In the context of the European project LUISA[4], we propose to move from the existing data and metadata based paradigm to a highly dynamic service-oriented approach based on semantic Web service technologies. To enable this vision, we adopt a semantic approach which abstract from data, services and existing process metadata standards. By making use of ontologies, we represent (i) a process in terms of sequences of learning goals to achieve and (ii) the learning context – i.e. domain requirements or learner profile and preferences - where the process is performed. At runtime, given a learning goal and the reference process context, IRS-III can identify and deliver the most appropriate resources that allow the learner to accomplish such a goal.

[4] http://kmi.open.ac.uk/projects/luisa/

As a result, we enhance the current state of the art by enabling context adaptive e-Learning applications based on distributed, flexible and open infrastructures. Starting from our semantic representation of processes, we can ground to multiple existing metadata standards and thus reuse the respective runtime environments. Instead of grounding the metadata standard activities to static learning resources, we link them to our learning goals. When the standard-compliant application processes an activity, the associated learning goal is invoked. Several services on top of repositories from different organizations can be linked to and thus provide resources for a specific learning goal. Their selection is based on axioms that declare the assumed learning context for a service. If a specific goal is not achievable by existing services, an opportune SWS orchestration can create on-the-fly integrated services. Note that for each goal new services can be easily integrated by simply introducing the respective semantic descriptions, without affecting the existing structure. Finally, each service can provide resource following different standards (or not following any standard at all), since appropriate mediation services can be used to address existing data heterogeneities.

### 3.3 E-Science

A number of large research initiatives[5] aim to develop computer models of human physiology that span multiple dimensional and temporal scales. EuroPhysiome[6] is an initiative to promote the development of the Virtual Physiological Human (VPH), a methodological and technological framework that will enable medical investigators to consider the human body as a single (though still hugely complex) system.

While the VPH will have a sizeable impact on all branches of biomedical research and clinical medicine, a primary target domain is the Musculoskeletal System, which we address in the Living Human Digital Library (LHDL[7]) project.

In LHDL, we serve a virtual community comprising students and professionals engaged in researching the musculoskeletal system. They are interested in accessing and managing complex biomedical data. Using Web services, LHDL researchers can share data, algorithms, and community services. In a large and complex domain like the biomedical one, understanding and coordinating these services is a major task. By adding formal semantic descriptions of the Web services, we can recruit computers to perform much of this reasoning for us.

IRS-III uses these semantic annotations to alleviate many of problems that usually impede full and easy interoperability between the kinds of heterogeneous resources deployed in huge context such as VPH. IRS-III assists in the technical tasks of finding, composing, and resolving mismatches between Web service components, as well as reason about high-level policy issues such as patient privacy, data security and provenance, and computational resourcing. In more detail, Web services represent the services that each VPH community will expose. A VPH user performs a request by a VPH portal; the portal sends the request which captures a desired outcome or goal

[5] http://www.mygrid.org.uk/, http://www.esnw.ac.uk/
[6] http://www.europhysiome.org/
[7] http://kmi.open.ac.uk/projects/lhdl/

and, using a set of semantic Web service descriptions, IRS-III will: a) discover potentially relevant Web services in any VPH sub-community; b) select the set of Web services which best satisfy the user request; c) mediate any mismatches; and d) invoke the selected VPH Web services according with any Web service invocation constraints.

The execution sequence of a complex semantic Web service is not hard-coded, but it is dynamically created using goal-based discovery and invocation: several Web services may be associated with a goal, and only the most applicable will be discovered and invoked at runtime (late binding). If a new service is available within one VPH community, the developers simply need to describe and then link it to an existing goal; if a service is altered, only the specific semantic description will be affected, and not the whole business process.

## 4 Conclusions

Semantic Web services research has the overall vision of bringing the Web to its full potential by enabling applications to be created automatically from available Web services in order to satisfy user goals. Fulfilling this vision will radically change the character of all online interaction including the nature of e-Commerce, e-Science, e-Learning, and e-Government. Key to achieving this vision is the provision of SWS platforms able to support the development and use of online libraries of reusable software components indexed through generic and domain specific ontologies. In this paper we have presented our SWS platform IRS-III, which contains a suite of tools to enable the development and management of semantic descriptions. Using the semantic Web service descriptions, IRS-III, through orchestration, mediation and choreography components, can broker between incoming goal requests and applicable Web services.

Over the past few years we have used IRS-III to create a number of SWS based applications. Within a number of new EU funded projects we are currently creating applications in the areas of: business process modelling, linking IRS-III to a BPEL engine[3]; e-learning, integrating IRS-III with learning resource repositories[4]; and, bio-informatics, describing Grid services related to the human musculo-skeletal system[7]. The diversity of the domains in which we are able to deploy IRS-III is evidence of the utility and robustness of our approach, and, we fully expect to gain further valuable insights into the overall requirements for semantic Web services during the deployment process. In this respect we welcome external parties to use our platform - the IRS-III API and browser for can be downloaded from the IRS-III Web site at http://kmi.open.ac.uk/projects/irs/.

**Acknowledgments.** This work was supported by the SUPER (Semantics Utilized for Process management within and between Enterprises) project, (FP6 – 026850); LUISA (Learning Content Management System Using Innovative Semantic Web Services Architecture) project (FP6 – 027149); LHDL (Living Human Digital Library) project (FP6 – 026932).

## References


1. Cabral, L., Domingue, J.: Mediation of Semantic Web Services in IRS-III. In Proceeding of the Workshop on Mediation in Semantic Web Services in conjunction with the 3rd International Conference on Service Oriented Computing, Amsterdam, The Netherlands, (2005).
2. Cabral, L., Domingue, J., Galizia, S., Gugliotta, A., Norton, B., Tanasescu, V., Pedrinaci, C.: IRS-III: A Broker for Semantic Web Services based Applications. In Proceedings of the 5th International Semantic Web Conference, Athens, USA, November, (2006).
3. Domingue, J., Cabral, L., Galizia, S., and Motta, E.: A Comprehensive Approach to Creating and Using Semantic Web Services, In Proceedings of the W3C Workshop on Frameworks for Semantics in Web Service, Innsbruck, Austria, June 9-10, (2005).
4. Domingue, J., Galizia, S., and Cabral, L.: Choreography in IRS-III- Coping with Heterogeneous Interaction Patterns in Web Services. In Proceedings of 4th International Semantic Web Conference, Galway, Ireland, (2005).
5. Fensel, D., Lausen, H., Polleres, A., De Bruijn, J., Stollberg, M., Roman, D., Domingue, J.: Enabling Semantic Web Services: Web Service Modeling Ontology. Springer, (2006).
6. Hepp, M., Leymann, F., Domingue, J., Wahler, A. and Fensel, D.: Semantic business process management: A vision towards using semantic web services for business process management. In ICEBE, pages 535–540, (2005).
7. Motta, E.: An Overview of the OCML Modelling Language, In Proceedings of the 8th Workshop on Knowledge Engineering Methods and Languages (KEML '98). (1998).
8. Motta, E., Domingue, J., Cabral, L., and Gaspari, M.: IRS-II: A Framework and Infrastructure for Semantic Web Services. In Proceedings of the 2nd International Semantic Web Conference (ISWC2003), 20-23 October 2003, Sanibel Island, Florida, USA.
9. OWL-S Working Group: OWL-S 1.2 Pre-Release, (http://www.ai.sri.com/daml/services/owl-s/1.2/). (2006).
10. SOAP: SOAP Version 1.2 Part 0: Primer, (http://www.w3.org/TR/soap12-part0/. (2003).
11. Stollberg, M. and Norton, B.: A Refined Goal Model for Semantic Web Services. In Proc. of the 2nd International Conference on Internet and Web Applications and Services (ICIW 2007), Mauritius, (2007).
12. UDDI: UDDI Spec Technical Committee Specification v. 3.0, http://uddi.org/pubs/uddi-v3.0.1-20031014.htm. (2003).
13. Van der Aalst, W., Ter Hofstede, A., and Weske, M.: Business process management: A survey. In Business Process Management, pages 1–12, (2003).
14. WSDL: Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/2001/NOTE-wsdl-20010315. (2001).