Mediation of Semantic Web Services in IRS-III

Liliana Cabral and John Domingue

Knowledge Media Institute, The Open University, Milton Keynes, UK
{L.S.Cabral, J.B.Domingue}@open.ac.uk

Abstract. Business applications composed of heterogeneous distributed components or Web services need mediation to resolve data and process mismatches at runtime. This paper describes mediation in IRS-III, a framework and platform for developing WSMO-based Semantic Web Services. We present our approach to mediation within Semantic Web Services and highlight the role of WSMO mediator types when solving mismatches at the semantic level between a service requester and a service provider. We describe the components of our mediation framework and how it can handle data, goal and process mediation during the activities of selection, composition and invocation of Semantic Web Services.

1 Introduction

Integrating software applications developed in heterogeneous platforms has a high cost for most businesses today, because it means manually providing mappings for data and message formats exchanged between business processes of partner agencies. The advent of Web Services, as part of a trend in XML-based distributed computing, made the integration of applications on the Web a far easier task. Companies can keep intact their legacy implementation of computing systems and provide services by exposing functionalities through a standard interface description. Thus, applications in diverse areas such as e-commerce and e-government can interoperate through Web services implemented in heterogeneous platforms.

However, integration requires that requesters of Web services agree on the meaning of the messages being exchanged with the providers before they can invoke the Web services. In addition, a service requester has to map his request to the requirements of available services.

Despite the advance in the use of standards for Web Service description (e.g. WSDL) and publishing (e.g. UDDI), the syntactic definitions used in these specifications do not completely describe the capability of a service and cannot be understood by software programs. It requires a human to interpret the meaning of inputs, outputs and applicable constraints as well the context in which services can be used. Semantic Web Services (SWS) combine Semantic Web and Web Service technologies to provide the infrastructure for semantically describing Web services facilitating automatic service discovery, composition, execution and mediation.

This paper describes mediation in IRS-III [6], an infrastructure for developing WSMO-based Semantic Web Services [17]. IRS-III is an operational semantic plat-

form for the representation and execution of knowledge models. We present our top-down approach to mediation within Semantic Web Services and highlight the role of WSMO mediator types when solving mismatches at the semantic level between a service requester and a service provider. We describe the components of our mediation framework and how it can handle data, goal and process mediation during the activities of selection, composition and invocation of Semantic Web Services.

The rest of the paper is structured as follows: section 2 describes mediation issues faced by applications using Semantic Web Services; section 3 gives a brief overview of IRS-III and the Web Service Modeling Ontology – WSMO; section 4 describes the IRS-III mediation approach and in particular our mediation framework; in section 5 we present a case-study on e-government, which uses our approach; in section 6 we discuss related work and in section 7 we present our conclusions.

2 Semantic Mediation Issues

Business applications composed of heterogeneous distributed components or Web Services need mediation to resolve data and process mismatches at runtime. We view mediation in the context of Semantic Web Services and define it as an activity for solving conceptual mismatches during the interaction between a service requester and a service provider. One can model specialized mediators which provide a mediation service or declarative mappings for solving different types of mismatches.

Providing a semantic description for a Web Service allows a broker to use the knowledge available for managing the different levels of mediation needed. In this case the conversation between a client and a provider can be handled by a Semantic Web Service execution environment (broker) which can provide mediation during the activities of discovery, composition and invocation for solving mismatches at the semantic level.

A Semantic Web Service can be associated with one or more domain ontologies for describing its functional and non-functional capabilities. This description is used whenever a service is queried or invoked. Usually, a mapping between elements of the ontology used by the client application (or another service) and the ontology used by the service has to be provided. In particular, a developer might want to represent the connections and transformations between elements representing different aspects of the service, for example, for supporting dataflow of composed services. It might be also necessary to transform inputs during the selection and invocation process.

We present our view on the levels of mediation needed within a Semantic Web Service infrastructure that can be handled by different mediation components as well as a specific approach for modeling mediators which can represent types of mismatches and provide the mappings needed.

Semantic data mediation tackles the problem of alignment between ontologies associated with data resources. This problem alone is one of the main research topics on ontology management and coordination (e.g. [2], [7], [10]) in the Semantic Web, which investigates solutions in terms of automatically or semi-automatically generating declarative mappings between different ontological elements.

Within a Semantic Web Services infrastructure, domain ontologies are associated with the descriptions of the different elements of the service. Mediators between ontologies can carry out mappings when other elements such as Goals and Web Services import ontologies.

When composing services for providing functionality, the connections between services must match. Explicit mediators can be defined for mapping or transforming the output of a source service into the input of a target service.

When an application or another service has to interact with a service during invocation, mismatches can occur for example between the format or the order in which the information is requested and the way in which information is provided. A communication protocol in terms of message exchanges has to be followed by the service requester.

The mediation issues mentioned above can be solved by a mediation framework as part of the Semantic Web Service infrastructure. Different components of the run-time environment have access to the semantic descriptions of the service and are able to solve existing mismatches.

A mediation framework can be supported by a design-time tool. The design-time tool should support users in generating conceptual mappings between ontologies associated with Semantic Web Services. These declarative mappings can be made available through Mediators to the run-time environment, which is able to execute them during the invocation of a Semantic Web Service. Alternatively, the run-time environment can consume a mediation service associated with a Mediator, which can perform generic types of transformations on behalf of the service, for instance concatenations or sorting.

3 IRS-III Overview

IRS-III [6] is an implemented infrastructure which allows the description, publication and execution of Semantic Web Services according to the WSMO conceptual model [17]. The meta-model of WSMO defines four top level elements:
- Ontologies,
- Goals,
- Web Services, and
- Mediators.

Ontologies [8] provide the foundation for semantically describing data in order to achieve semantic interoperability and are used by the three other WSMO components. Goals define the tasks that a service requester expects a Web Service to fulfil. In this sense they express the service requester's intent. Web Service descriptions represent the functional behaviour of an existing deployed Web Service. The description also outlines how Web Services communicate (choreography) and how they are composed (orchestration). Mediators handle data and process interoperability issues that arise when handling heterogeneous systems. One of the main characterizing features of WSMO is that Ontologies, Goals and Web Services are linked by Mediators. In particular, WSMO provides four kinds of mediators:
- **OO-mediators** enable components to import heterogeneous ontologies;

- **WW-mediators** link Web Services to Web Services;
- **WG-mediators** connect Web Services with Goals;
- **GG-mediators** link different Goals.

The incorporation of four classes of mediators in WSMO facilitates the clean separation of different mapping mechanisms. For example, an OO-mediator may specify an ontology mapping between two ontologies whereas a GG-mediator may specify a process transformation between two Goals.

IRS-III provides a powerful execution environment for knowledge models. A WSMO description representing the capability of a deployed service can be instantiated within IRS-III operational framework and used for discovery, composition and invocation. IRS-III is based on a distributed architecture which communicates via SOAP. The server component handles ontology management and the execution of knowledge models for Semantic Web Services. The server also receives SOAP requests (through the API) from client applications for creating and editing WSMO descriptions of Goals, Services and Mediators as well as invocation of Goals. The publisher component allows providers of services to attach WSMO descriptions to their deployed services and provides handlers (proxies) to invoke services in specific platforms (lisp code, java code, web services and web applications).

4 IRS-III Mediation Approach

IRS-III Mediation approach is based on: a set of design principles for Semantic Web Services; a mediation framework incorporating a number of components of the IRS-III architecture; and use of the WSMO Mediator meta-model. The following subsections explain our approach in details.

4.1 Design Principles

Our approach is based on the following design principles:

Use of Ontologies – Semantic descriptions of Web Services are ontological meta-models. Furthermore, ontologies can serve as a container and delimit the scope of instances during the execution of a model for mediation.

Executable Semantic Descriptions – All aspects of a web service needed for mediation including choreography, orchestration and ontology mappings can be interpreted by the IRS-III execution environment (reasoning engine).

IRS-III as a broker – IRS-III mediates between client requests and service providers whenever a Semantic Web Service is invoked. The interaction with the service occurs via the choreography of a single service or via the orchestration of a composed service (multiple choreographies).

Goal-based invocation – Client requests are given by Goal descriptions. IRS-III selects a Web-Service which can achieve the Goal. Mismatches can occur between the Goal description and the Web Service description.

Goal-based decomposition – A Web-Service can decompose its functionality into sub-goals described by the orchestration. There can be mismatches between sub-Goals.

Explicit Semantic Mediator description – IRS-III uses mediators to explicitly connect and provide mapping rules or mediation services between services elements.

4.2 Mediation Framework

The IRS-III mediation framework implements data mediation, goal mediation and process mediation of Semantic Web Services. The main objective is to provide mediation components which solve types of mismatches by reasoning over the given Goal, Web Service and Mediator descriptions.

The following sub-sections will explain in more details the use of WSMO conceptual models by the *data mediator, goal mediator* and *process mediator* framework components.
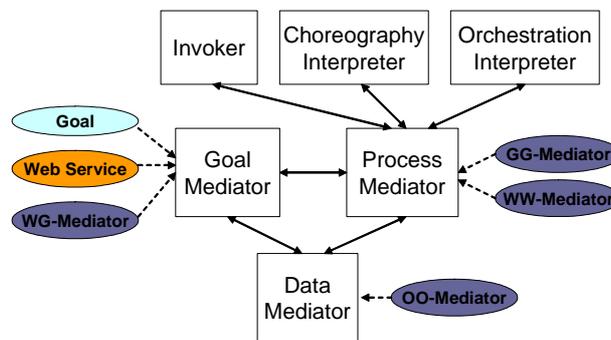
Fig. 1. Mediation framework of IRS-III

Figure 1 illustrates the main architecture components incorporated in the mediation framework of IRS-III. In the steps below we describe the overall sequence of mediation activities taking place during selection, composition and invocation of Semantic Web Services.

1. The Goal Mediator searches for WG-Mediators whose source component matches the current Goal when IRS-III receives an achieve-goal request from a client application. It selects the first targeted Web-Service which matches the requested capabilities (input types, preconditions, assumptions, non-functional properties etc). The types of mismatches that can occur are: a) the input types of a Goal are different from the input types of the target Web Service; and b) Web Services have more inputs than the Goal.

2. The Process Mediator establishes an interaction with a deployed web service (code) by executing its Web Service choreography through the Choreography Interpreter. The Process Mediator performs the lifting and lowering of data provided by the Web Service grounding and is able to create the communication messages corresponding to the choreography communication primitives.

It keeps the state of the communication throughout operation calls executed via the Invoker component.

3. The Process Mediator component also executes the orchestration of a composite Web Service using the Orchestration Interpreter. It keeps the state of the orchestration (control and data flow) between invocations of sub-Goals. The Process Mediator searches for GG-mediators connecting sub-Goals in the orchestration. The types of mismatches that can occur are: a) output types of a sub-goal are different from the input types of the target sub-Goal; b) output values of a sub-goal are in a different order from the inputs of the target sub-Goal; c) the output of a sub-Goal has to be split or concatenated into the inputs of the target sub-goals.

4. The Data Mediator component is used by the Goal Mediator and by the Process Mediator for mapping data across domain ontologies. It executes the mapping rules of OO-mediators used by other WSMO elements.

As a knowledge-based framework, IRS-III models the WSMO specification as a set of related knowledge models for the WSMO top level components of Goals, Web Services and Mediators, which are meta-models in corresponding ontologies. In the following we describe data, goal and process mediation from the perspective of the given Mediator model.

Listing 1. IRS-III mediator meta-model.

```
(def-class mediator (invokable-entity wsmo-entity)
  ((has-source-component :type wsmo-entity)
   (has-target-component :type wsmo-entity)))

(def-class wg-mediator (mediator)
  ((has-source-component :type (or web-service goal wg-mediator))
   (has-target-component :type (or web-service goal wg-mediator))
   (uses-mediator :type oo-mediator)
   (has-mediation-service :type goal)))

(def-class ww-mediator (mediator)
  ((has-source-component :type (or web-service ww-mediator))
   (has-target-component :type (or web-service ww-mediator))
   (uses-mediator :type oo-mediator)
   (has-mediation-service :type goal)))

(def-class gg-mediator (mediator)
  ((has-source-component :type (or goal gg-mediator))
   (has-target-component :type (or goal gg-mediator))
   (uses-mediator :type oo-mediator)
   (has-mediation-service :type goal)))

(def-class oo-mediator (mediator)
  ((has-source-component :type wsmo-ontology)
   (has-target-component :type wsmo-ontology
   (has-mapping-rules :type mapping-rule)))
```

Listing 1 shows the meta-model specification of a Mediator in IRS-III. The main concept is defined by the class *Mediator* which is subclassed into more specific types of mediators (wg-mediator, ww-mediator, gg-mediator, oo-mediator). Source and target components can be any of the WSMO top level components (class *wsmo-entity*). The mediators differ according to the type of source and target components they can

handle and whether it uses a mediation service or mapping rules. Thus, mediators are bridges which can provide conceptual mappings or input transformations from source components to target components. IRS-III supports the implementation of Mediation Services as Goals as well as the explicit declaration of mapping rules. Since mediation services are implemented as Goals they can simply be invoked resulting in the transformation of the relevant input data. IRS-III's reasoning engine can for example match the inputs of the mediation service with the inputs of the source component; and the output of the mediation service with the input of the target component.

4.2.1 Data Mediation

The Goal, Web Service and other Mediator descriptions associated with a web service can refer to an OO-mediator in order to use ontologies which do not match. IRS-III handles data mediation by executing the mapping rules provided by an OO-Mediator (fig. 2). In IRS-III, the source and target components of an OO-mediator are ontologies.. Furthermore, the source and target can be the home ontologies of associated Goals or Web Services.
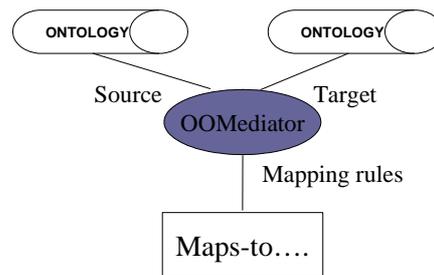

Fig. 2. Mediation between two ontologies

The underlying modeling language OCML [12] has a mechanism for mapping between entities associated with knowledge models. A simple way of dynamically associating elements of a source and a target ontology is through backward chaining rules using the *def-concept-mapping* and *def-relation-mapping* constructs.

Listing 2 shows how a mapping rule can be used to link the slots of classes in two different ontologies. More specifically, the definitions below link the *has-citizen-name* slot of class *citizen* in the source ontology to the *has-client-name* slot of class *client* in the target ontology. This example is taken from the e-government scenario and reflects the fact that a service requester can refer to an entity as citizen and a service provider can refer to it as client.

The *def-concept-mapping* construct in Listing 2 associates each instance of the *citizen* class to a newly created instance of the *client* class and link them by generating instances of the relation *maps-to* internally. The *def-relation-mapping* construct uses the generated *maps-to* relation within a rule which asserts the value of the mapped citizen name to the value of the client name.

IRS-III executes the mapping rules within a temporary ontology created by merging the source and target ontologies. The temporary ontology is then discarded after the Web Service invocation.

Listing 2. Example of a mapping rule.

```
(def-concept-mapping citizen client)

(def-relation-mapping citizen-client-name-mapping
   ((has-client-name ?client ?value)
    if
    (maps-to ?client ?citizen)
    (has-citizen-name ?citizen ?value)))
```

WG-mediators, GG-mediators and WW-mediators have a data mediation capacity for transforming inputs between source and target components by using mediation services and have different roles within the process mediation as explained in the following sections.

4.2.2 Goal Mediation

The goal mediator component of IRS-III handles mismatches that occur during the process of selection of Web Services for solving a Goal. The IRS-III approach assumes that application developers can create or search for Goal and Web Service descriptions available in a library.

A WG-mediator is created for connecting every Web Service to a Goal it can achieve. The WG-mediator model also specifies a mediation service which can transform the inputs of a Goal into the format of the inputs used by a Web Service. When a user requests a goal to be achieved, the mediation service associated with the mediator of each linked web service is executed so that the matchmaking during selection can be carried out over the mediated data.
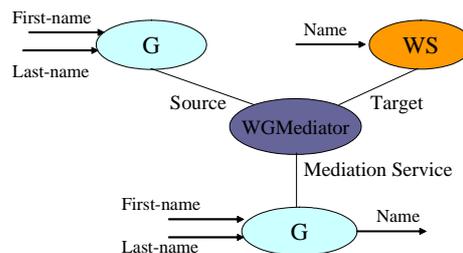

Fig. 3a. Mediation between a Goal and a Web Service. Two inputs of Goal are transformed into one input of the Web Service

Figure 3a shows a graphical illustration of the mediation taking place between a Goal and a Web Service via a WG-mediator. In this example, the Goal requested by the application takes two inputs (first and last names), which are transformed by the mediation service into one input (name) used by the target Web Service.
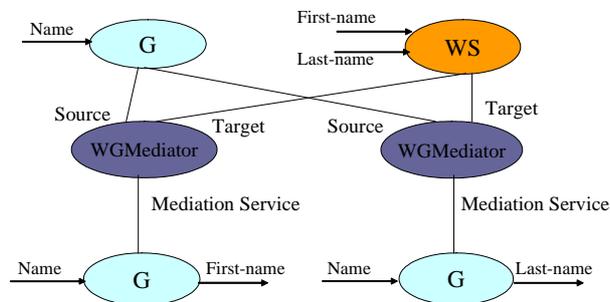
Fig. 3b. Mediation between a Goal and Web Service. One input of the Goal is transformed in 2 inputs of the Web Service

Since a mediation service can return only one output, IRS-III use a set of mediators between the goal (source) and the web service (target) in order to provide the required number of inputs to the target component as shown in figure 3b. In this example, each mediation service transforms (e.g. splits) the goal input (name) in one of the required inputs of the target component (first-name, last-name). The IRS-III engine can match the inputs and outputs for providing values as required.

4.2.3 Process Mediation

The Process mediation component of IRS-III handles mismatches that occur during the invocation or composition of a Web Service. IRS-III either executes the choreography (interaction protocol) of a single Web Service or the orchestration of a composed Web Service, using the values provided by the Goal inputs. Moreover, the Process mediator has to execute the choreography of each single Web Service in the Orchestration.

In IRS-III the choreography of a Web Service describes how to interact with a deployed service (client choreography). A set of rules (guarded transitions) in the choreography are used to specify the flow of operations required for realizing the specific functionality of the Web Service. The Process Mediator uses the Web Service grounding for creating the communication messages based on the operations declared at the conceptual level.

A choreography is provided to interact with a single Web Service. By interpreting the choreography and grounding, the Process mediator component can send messages to the service in the right order and format on behalf of the client. When a Web Service is composite an orchestration has to be provided instead. Nevertheless, its input values have to be passed to the orchestration and the result of the orchestration has to be passed back to the Web Service. The orchestration follows the decomposition of Goals into sub-Goals and uses GG-mediators for connecting sub-goals and mediating the order and types of inputs between them.

We illustrate in the following the role of a GG-mediator during orchestration (fig. 4). The provider of a Web service describes the orchestration through control-flow mechanisms, for instance: (sequence G1 G2 M1). The *Sequence* control command

executes the given sub-goals (G1 and G2) in sequence. Figure 4 shows the graphical representation of the GG-mediator connecting G1 to G2. This mediator supports the data flow between the sub-goals and the necessary transformations. The source goal (G1) produces one output (E1), which is transformed by the mediation service in one input (E2) used by the target Goal (G2). During the execution of the orchestration the input values (SC, TC, A) received by the current invoked Goal are sent onto the sub-goals through matching, then the associated GG-mediator (M1) are used to connect and forward results between sub-goals providing the necessary transformations through the mediation service.
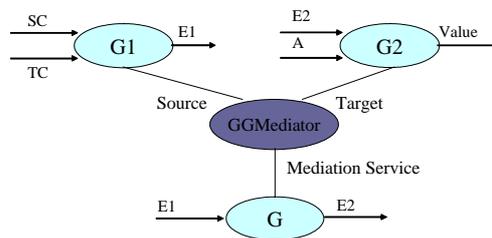

Fig. 4. Mediation between two sub-goals. The input of goal1 is transformed in one input of goal2

WW-mediators can be used in a similar way to GG-mediators by the Process Mediator. In this case, the WW-mediator can provide mappings between the input values of the current Web Service and the Web Services in the orchestration.

5 A case study in E-government

The main requirement for applications in E-government relates to the interoperability of data and processes between services provided by government agencies. Thus, the e-government domain is a natural application area for mediation of Semantic Web Services. The ability to aggregate and re-use all the information resources relevant to a given problem and further to make this available as a basis for transparent interaction with community partner organisations and individual citizens is very restricted. Furthermore, the goals of citizens using e-government services and of government providers of services are often not conceptually aligned, contributing to misunderstanding, low take up and poor relations between citizens and their governments.

We have created a prototype for the case study on e-government within the DIP project (http://dip.semanticweb.org) for illustrating Semantic Web Services. We will comment on the requirements and use of mediation within the scenario implemented.

5.1 Application Scenario and requirements

We illustrate the implementation of our e-government use case through an application scenario. The prototype is a portal for Essex County Council in UK, where two governmental agencies were involved:

- **Community Care (Social Services) in Essex County Council** - they typically have a coordinating role in relation to a range of services from a number of providers and special responsibility for key services such as support for elderly and disabled people (day centers, transportation). It uses the SWIFT database as its main records management tool.

- **The Housing Department of Chelmsford District Council** - handles housing services and uses the ELMS database

In this scenario, a case worker of the Community Care department helps a citizen to report his/her change of circumstance (e.g. address) to different agencies involved in the process. In this way, the citizen only has to inform the council once about his/her change, and the government agency automatically notifies all the agencies involved. An example might be when a disabled mother moves into her daughter's home. The case worker opens a case for a citizen who is eligible to receive services and benefits – health, housing, etc. Multiple service providing agencies need to be informed and interact.

From the scenario above we have gathered the following mediation requirements and solutions:

- Data mediation - Agencies have their own databases and hence different data formats for the same concept (e.g. Address). Different data formats can be lifted to the same concept in a domain ontology. At a semantic level, different concepts can be mapped through mediators.
- Goal mediation – Agencies achieve goals in different ways (e.g assess equipment for a citizen). Here we can define one Goal that can be satisfied in different ways by applicable Web Services developed within different agencies. Multiple Web Services can be linked to the same Goal via Mediators.
- Process mediation - Agencies processes behave in different ways according to their own set of operational procedures, requirements and constraints. Each Web Service presents a choreography describing how a client talks to the deployed service. Furthermore, sub-Goals can be composed together for providing the functionality of one Web Service through the Web Service orchestration.

5.2 Prototype Development

In our approach for developing applications using Semantic Web Services with IRS-III we devise a customer team for creating Goal descriptions according to user requests and a development team for creating Web Service descriptions for the available deployed web services. The application developer is then able to create Mediator descriptions which connect domain ontologies, Goals and Web Services and provide

mediation services or mapping rules for solving mismatches between ontological elements.

The main characteristic of the prototype architecture is that it is service oriented. The portal application created over this architecture implements Semantic Web Services that have integration purpose across the various agencies involved in our e-government scenario. The main services provided through the portal are the ones which can be shared between agencies or used to send/get information to/from more than one agency or even third parties (e.g. list of equipments provided).

The structure of ontologies in fig. 5 represents the libraries of WSMO models for the e-government application. The light-colored rectangles on the top-half of the diagram represent domain ontologies. The dark-colored rectangles on the bottom-half of the diagram represent ontologies with Goal, Web Service and Mediator descriptions available from Community Care (boxes on the left) and Housing Department (boxes on the right. The libraries above provide a clear separation of user goals and web service contexts within agencies and the use of mediators for linking them. Agencies also share the WSMO upper model and the e-government upper level ontology.
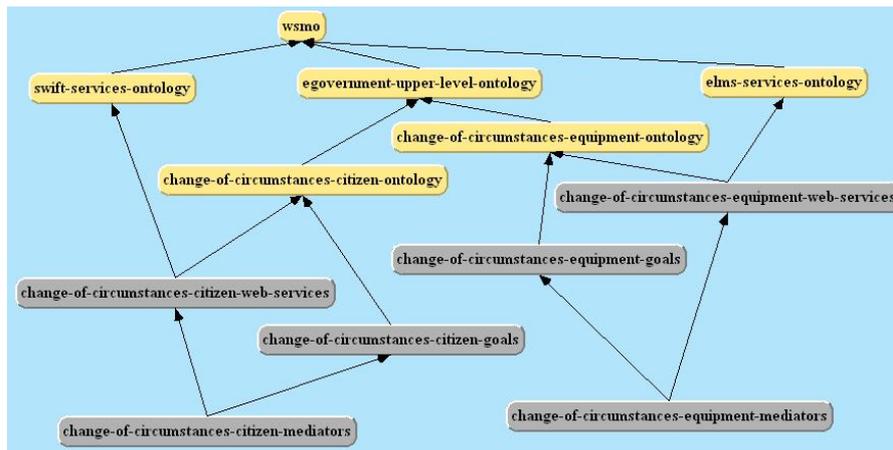

Fig. 5. Structure of ontologies for the e-government application

For illustration purposes (figure 6) we describe in the following the structure of WSMO descriptions associated with one of the goals (*Assess-Equipment-Goal*) defined in our prototype. This Goal describes a request for a service that can assess housing equipments (items) for a citizen who has registered for benefits within Essex County Council. Published services must find all items that suit the citizen's situation (mobility-impairment, visual-impairment, hearing-impairment, baby-care etc) and weight, and the budget of the council's case worker. Restrictions on the way the service can solve the goal are given by pre-conditions and post-conditions.

The Housing Department provides a composed web service (*Housing-Dept-Assess-Items-WS)* for solving the goal described above. The composition is defined by the orchestration in the format: *(Sequence G1 G2 G3 M1 M2),* where G1, G2 and G3 represent sub-goals and M1 and M2 the GG-mediators connecting them. In our example

the sub-goals are: *Find-Items-by-Purpose-and-Weight-Goal*; *Assess-Budget-Goal* and *Select-Suitable-Items-Goal*. A third party company provides a single web service (*Third-Party-Assess-Items-WS)* for solving the above goal, which is described with concepts from the domain ontology *Third-Party-Items-Ontology*.

The mediator descriptions used in this example (fig. 6) are explained in the following. Note that all links coming from mediators connect source to target components (labels were omitted to avoid cluttering the diagram).
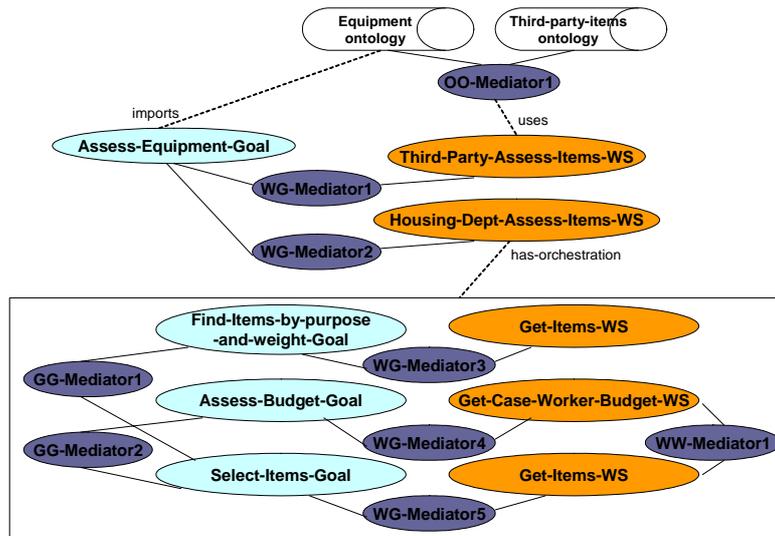

Fig. 6. Sample structure of WSMO descriptions for the e-government prototype

- **WG-Mediator1** – connects *Third-Party-Assess-Items-WS* to E-Gov-Assess-Items-Goal allowing it to be selected for solving the goal. This mediator defines a mediation service for converting the value of input *weight* from pounds (in the goal) to kilos (in the web-service).
- **WG-Mediator2** – connects *Housing-Dept-Assess-Items-WS* to E-Gov-Assess-Items-Goal allowing it to be selected for solving the goal. There is no mediation service and the input types are inherited from the goal.
- **OO-Mediator1** – Defines mapping rules for aligning *Third-Party-Items* domain ontology (used by the Web Service) with Equipment ontology;
- **GG-Mediator1** – Allows the output of *Find-Items-by-Purpose-and-Weight-Goal* to be used as input by *Select-Suitable-Items-Goal*.
- **GG-Mediator2** – Allows the output of *Assess-Budget-Goal* to be used as input by *Select-Suitable-Items-Goal*. It uses a mediation service to map the input type *Budget* (in the source sub-goal) to input type *Cost* (in the target sub-goal).
- **WG-Mediator3**, **WG-Mediator4** and **WG-Mediator5** – Connect corresponding Web Services to Sub-Goals in the orchestration. The Housing

Department has specific services for solving those goals and no mappings are required.

- **WW-Mediator1** – Connects the two web-services for sharing concepts.

6 Discussion and Related Work

Mediation approaches for integration of heterogeneous components or data resources can range from techniques for mapping several resources to a canonical ontology to mediation components which handle transformations between different protocols. In this paper we focus on mediation provided by Semantic Web Services and in particular, on the conceptual modeling and integration aspects of mediation rather than on mapping algorithms. Thus, we have investigated how a mediation framework can handle semantic descriptions for solving mismatches during selection, composition and invocation of services.

Recent work within the knowledge representation research community (e.g. [9] [16] [10]) has contributed to the formalization of ontological mappings, which can be used by SWS mediators, specifically OO-mediators. Reuse in ontology mappings is also discussed in [7], where types of mappings between ontologies, called alignments, are viewed as objects which can be created and used by different users. However, in that case the API proposed is more likely to be used within a design tool which would generate the mappings declaration.

OWL-S [15] does not model the mediator concept. Yet, mediation plays a key role in the approach [14]. The OWL-S approach considers that mediation is handled during discovery and decomposition by architectural components and that a mediation service is treated just as another web service. This assumption makes mediation very implementation dependent and not visible to the user.

WSMX [18], an execution environment for WSMO, contains a data mediation component [11] and a process mediation component [4]. The main difference is that WSMX is not a knowledge-based execution environment. Thus, the Mediator conceptual model is not used by the Mediation components. The WSMX data mediation component can execute mapping rules generated at design time by a mapping tool, but do not implement mediation services as Goals. The WSMX process mediation component works on predefined types of mismatches between two choreography instances while IRS-III Process Mediator interprets the choreography provided by the Web Service and handles mismatches during orchestration. IRS-III follows the UPML design principles [13] for Goal decomposition within the orchestration, whereby a Web Service can decompose its functionality into sub-Goals.

The work on virtual providers [1], which formalizes WSMX process mediation, follows the same approach for mediating between two business process interfaces. The approach in [5] describes the process mediator as the middleware for handling composition.

7 Summary and Conclusions

In this paper we have identified mediation issues within Semantic Web Services and provided a solution in terms of a mediation framework. IRS-III as a broker follows a top-down approach in which its framework components use semantic descriptions to support the mediation process. IRS-III enables easy integration of heterogeneous services by reasoning over the WSMO conceptual model of domain ontologies, Goals, Web Services and Mediators. In particular, modeling mediators provides design time and runtime support for the automation of data, goal and process mediation of Semantic Web Services.

Explicitly modeling types of Mediators in WSMO has many advantages since IRS-III provides the mechanisms for reasoning and behaving according to the knowledge models. First, representing a mediator as a meta-model enables easy inspection by developers. Second, as independent components, mediators can be indexed and reused through a library. Third, mediation services associated with mediators are defined as goals which can be achieved by an implemented web service. Finally, ontology mappings can be provided by the relation mapping mechanism of the underlying reasoning engine. Developers can also inspect mapping rules and test mediation services by searching a library or repository of mediators.

IRS-III exploits the semantics of the WSMO mediator concept during the selection, composition and invocation of Semantic Web Services. For example, the IRS-III broker can match the inputs of the source component with the inputs of the mediation service when deciding which values will be mediated. In a similar fashion, mediators are used for dataflow between sub-goals during orchestration.

We have presented a case study on e-government, which offers a motivating scenario for the use of mediators. Further work is under development regarding the development of a design tool for generating mapping rules for mediators.

Acknowledgements

References

1. Altenhofen, M., Borger, E., Lemcke, J. (2005). A High-level specification for mediators. Accepted at 1st International Workshop on Web Service Choreography and Orchestration for Business Process Management.
2. Bouquet, P., Serafini, L. and Zanobini, S. (2004). Coordinating Semantic Peers. In proceedings of the 11th Int. Conf. AIMSA 2004, Varna, Bulgaria. LNAI 3192

3. Cabral, L., Domingue, J., Motta, E., Payne, T. and Hakimpour, F. (2004). Approaches to Semantic Web Services: An Overview and Comparisons. In proceedings of the First European Semantic Web Symposium, ESWS 2004, Heraklion, Crete, Greece. LNCS 3053, pp. 225-239

4. Cimpian, E.; Mocan, A. (2005): *WSMX Process Mediation Based on Choreographies*. 1st International Workshop on Web Service Choreography and Orchestration for Business Process Management, 2005.

5. Constantinescu, I., Faltings, B., Binder, W. (2004). Large scale, type-compatible service composition. In IEEE International Conference on Web Services (ICWS 2004, San Diego, CA, USA.

6. Domingue, L. Cabral, F. Hakimpour,D. Sell and E. Motta (2004). IRS-III: A Platform and Infrastructure for Creating WSMO-based Semantic Web Services. Proceedings of the Workshop on WSMO Implementations (WIW 2004) Frankfurt, Germany. CEUR Workshop Proceedings, Vol. 113

7. Euzenat, J. (2004). An API for Ontology Alignment. In proceedings of ISWC 2004, Third International Semantic Web Conference, Hiroshima, Japan. LNCS 3298.

8. Gruber, T. R. (1993) A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2)

9. Heller, B. and Herre, H. (2004). The Theory of Top-level Ontological Mappings and its Application to Clinical Trial Protocols. In proceedings of the 14th Int. Conf. EKAW 2004, Whittlebury Hall, UK. LNAI 3257

10. Kotis, K., Vouros, G. and Stergiou, K. (2004). Capturing Semantics Towards Automatic Coordination of Domain Ontologies. In proceedings of the 11th Int. Conf. AIMSA 2004, Varna, Bulgaria. LNAI 3192

11. Mocan, A.; Cimpian, E. (2005): *D13.3v0.2 WSMX Data Mediation*, WSMX Working Draft 2005, available at: http://www.wsmo.org/TR/d13/d13.3/v0.2/

12. Motta, E. (2004) Reusable Components for Knowledge Modelling: Case Studies in Parametric Design Problem Solving. IOSPress.

13. Omelayenko, B., Crubezy, M., Fensel, D., Benjamins, R., Wielinga, B., Motta, E., Musen, M., Ding, Y. (2003). UPML: The language and Tool Support for Makiing the Semantic Web Alive. In: Fensel, D. et al. (eds.): Spinning the Semantic Web: Bringing the WWW to its Full Potential. MIT Press 141–170

14. Paolucci. M, Srinivasan, N. and Sycara, K (2004). Expressing WSMO Mediators in OWL-S. In proceedings of the Workshop on Semantic Web Services: Preparing to meet the World of Business Applications, in conjunction with ISWC 2004, Hiroshima, Japan. http://CEUR-WS.org/Vol-119/

15. OWL-S Coalition (2003). OWL-S 1.0 Release. http://www.daml.org/services/owl-s/1.0/.

16. Van Elst, L. and Kiesel, M. (2004). Generating and Integrating Evidence for ontology mappings. In proceedings of the 14th Int. Conf. EKAW 2004, Whittlebury Hall, UK. LNAI 3257

17. WSMO Working Group (2004). D2v1.0. Web Service Modeling Ontology (WSMO). WSMO Working Draft. http://www.wsmo.org/2004/d2/v1.0/

18. WSMO Working Group (2004). D13.3v0.1 Mediation in WSMX. WSMX Working Draft. http://www.wsmo.org/2004/d13/d13.3/v0.1/