Mediating Capabilities with Delta-Relations

Michael Stollberg[1], Emilia Cimpian[2], and Dieter Fensel[1,2]

[1] Digital Enterprise Research Institute Innsbruck (DERI Austria), Institute for Computer Science, University of Innsbruck, Technikerstrasse 21a, A-6020 Innsbruck, Austria

[2] Digital Enterprise Research Institute (DERI Ireland), IDA Business Park, Lower Dangan, Galway, Ireland

Abstract. Mediation is concerned with handling heterogeneities that potentially occur between resources that shall interoperate. Heterogeneity being an inherent characteristic of open and distributed environments like the Internet, mediation becomes a core issue for next generation Web technologies. Recent developments around the Semantic Web and Semantic Web services address mediation on the data level and on the process level. This paper identifies the *teleological level* as a novel level of mediation that deals with heterogeneities of capabilities as the functional descriptions of Web services and service requests. The central mediation technique therefore are so-called Δ-relations that explicitly denote the logical relationships between capabilities. These can be used to perform central reasoning tasks for Semantic Web Services by simple inferences instead of more complex reasoning procedures, hence allow gaining efficiency in Semantic Web service technologies.

1 Introduction

The initial Web service technology stack around SOAP, WSDL, and UDDI remains on a syntactic level for describing Web services that limits Web service usage to manual inspection and integration. As this is considered to fail as a basis for dynamic service-oriented architectures, the emerging discipline of Semantic Web services develops semantically enabled technologies for automated discovery, composition, communication, cooperation, and execution of Web services. On basis of exhaustive semantic description frameworks and usage of ontologies as the underlying data model, Semantic Web services strive towards an integrated technology for realizing the vision of the Semantic Web [19], [6].

Apart from enabling advanced techniques for automated Web service usage, a main merit of Semantic Web service technology is the inherent support for handling heterogeneities on a semantic level [8]. Heterogeneity is an inherent characteristic of the Internet that hampers successful and efficient inter-operation of Web services, requests, and other resources. Semantic resource descriptions allow utilization of semantically enabled techniques for depicting and resolving heterogeneities. This is commonly referred to as mediation, wherefore different levels are distinguished with respect to the type of heterogeneities that can occur and techniques used for handling these.

The most prominent frameworks for Semantic Web services address mediation as follows. OWL-S [18] defines an ontology for semantically describing Web services that is comprised of Service Profiles, Service Models, and Grounding as top-level elements. Mediation is not considered as a central element but as an architectural aspect arising in concrete Web service systems. However, the intention of OWL-S is to provide a semantic description model for Web services while leaving technology development open to respective efforts; following this, OWL-S is declared to be orthogonal to mediation [23]. In contrast, the Web Service Modeling Ontology WSMO [16] depicts mediation as an integral aspect and hence defines an architectural model for mediators along with techniques for handling heterogeneities at different levels.

Ongoing research and development efforts on mediation address the data level including heterogeneities on terminologies and representation formats, and the process level as mismatches in Web service communication and cooperation. This paper identifies an additional level of mediation that deals with heterogeneities between functional descriptions of Web services and requests. Arising from decoupled and decentralized development, these might not hamper Web service usage but can cause significant decelerations in automated Web service technologies. This can be overcome by explicitly defining the logical relationship between functional descriptions that allow replacing complex procedures for central reasoning tasks by simpler inferences. We refer to this as the *teleological level of mediation* wherefore this paper presents so-called Δ-relations as the main mediation technique and usage in mediators for gaining efficiency in Semantic Web service technology.

This paper is structured as follows: Section 2 recalls the state of affairs in mediation techniques developed for Semantic Web services and introduces the teleological level of mediation; Section 3 presents the definition of Δ-relations and their usage in mediators; Section 4 exposes the benefits of teleological level mediation within exemplary scenarios; Section 5 discusses related work, and Section 6 concludes the paper.

2 Mediation Levels and Techniques

In order to motivate the new level of mediation introduced in this paper, the following recalls the need for mediation within Semantic Web services and depicts the potential benefits and requirements for teleological level mediation.

Semantic Web services aims at developing integrated technology for automating the complete Web service usage process [12]. This consists of the following, possibly iterative steps wherefore semantically enabled mechanisms are developed. At first, create Web service implementations along with their semantic description and make them accessible (*publication*), then detect appropriate Web services for solving a given request (*discovery*) or combine several services therefore (*composition*), choose the most adequate Web service out of the applicable ones for a request (*selection*), and finally access the chosen Web service (*invocation*) and control the information interchange for completing the service usage (*conversation*). Orthogonal to this, heterogeneities might occur that hamper automated Web services usage. Handling these is subject to mediation that becomes a major concern in Semantic Web and Semantic Web service technologies with regard to the open and decentralized nature of the Web.

Hence, architectural models for Semantic Web services are proclaimed that treat mediation as a first class citizen [8]. Following [25], a mediator is an architectural component capable of establishing interoperability of resources if not given a priori by resolving heterogeneities. Aiming at generic, domain and application independent mediation facilities, mediation techniques are envisioned that work declarative resource descriptions for detecting and handling heterogeneity on the semantic level. The ultimate aim of an integrated mediation framework as aspired in WSMO [20] is to provide means for handling and resolving all kinds of heterogeneities that might hamper Web service usage. Therefore, potentially occurring heterogeneities are classified into different levels with respect to the distinct mediation techniques and architectural requirements needed for handling these. The following inspects heterogeneity types and respective mediation levels along with recent developments and then reveals the necessity for teleological level mediation.

2.1 Data and Process Level Mediation

One kind of heterogeneity that can occur is usage of different terminologies by entities that shall interchange information. Within ontology-based environments like Semantic Web services, this means that heterogeneous ontologies are used as the terminological basis for element descriptions which hinders prosperous information interchange. Mediation techniques for handling terminological mismatches on an ontological level are ontology mapping, merging, and alignment, collectively referred to as ontology integration techniques [1]. Mismatches that hamper information interchange can also result from usage of different data representation formats or technical transfer protocols. A suitable way of resolving such heterogeneities is to lift the data from the syntactic to an ontological level, resolve the mismatches on this level, and then lower them again to the syntactic representation [22]. Because these two types of heterogeneity can be handled by similar techniques, they are consolidated as *data level mediation* [21].

Another type of heterogeneity can occur on the behavioral level that hamper entities from interacting successfully with respect to their individual business processes. For instance, at some point during the interaction of a requester R with a Web service S, R expects an acknowledgement while S waits for the next input; so, the interaction process between R and S runs into a deadlock situation. Such heterogeneities can be resolved by inspecting the individual business processes of the entities that shall interact and establish a valid process for interaction on basis of generic mediation operations on business processes. This is referred to as *process level mediation* [4].

The need for data and process level mediation for establishing interoperability of Web services and related elements if not given a priori has already been indicated in [8].

2.2 Functional Heterogeneity

Realization of Semantic Web technology reveals another type of heterogeneity that occurs as functional differences between Web services and requests. With respect to the distributed nature of the Web and the accompanying dispersed description of Web services and requests, the usual case is that the requested functionality does not precisely match with the one provided by a Web service. If we would have additional information on the relationship between functional descriptions, we could use this for improving the efficiency of semantic matchmaking components.

Consider the following example that we examine in Section 4 in more detail. There is a request R of 'finding information on Italian restaurants in Innsbruck' and some available Web services: WS_1 offers a hotel and restaurant guide for Innsbruck, WS_2 is a Tyrol restaurant guide, and WS_3 provides information on restaurants with traditional Tyrolean cuisine in Innsbruck. For detecting the Web services WS_1 and WS_2 to be usable for achieving R, we need to run a discovery procedure between R and each service. Existing developments for discovery with semantic matchmaking consist of complex, expensive reasoning procedures (see [17], [13], [15], [24]).

This effort can significantly be reduced when knowing and considering the relationship between R and the Web Services. Imagine that we have the following additional information: R is a specialization of another request R_O for finding information on restaurants, and from a previous discovery run, we know WS_1, WS_2, and WS_3 are usable for resolving R_O. If we compute Δ_{R,R_O} as the logical relationship between the requests, we can determine the Web services usable for resolving R as those usable for resolving R_O that also satisfy Δ_{R,R_O}. Hence, we can replace costly discovery runs for determining usable Web services for R by a much easier, straightforward, more efficient inference.

This is the motivation and aim of what we refer to as *teleological level mediation.* Heterogeneities as in the example appear as differences between functional descriptions, i.e. between OWL-S service profiles or WSMO capabilities. Specifying the functionality of a service or the one required for satisfying a request, these descriptions are concerned with the application purpose of Web services which can be referred to as the teleological level of Semantic Web service description ontologies. Knowing the explicit differences between functional descriptions of Web services and requests allows increasing the efficiency of Semantic Web service technologies by replacing complex reasoning tasks by simpler ones throughout the Web service usage process.

3 Teleological Level Mediation Techniques

With regard to the above examinations, the following introduces techniques for teleological level mediation. The basis of our approach are so-called Δ-relations that explicitly denote the logical relationship between functional resource descriptions. We provide the definition of Δ-relations, expose their beneficial usage, and integrate them into the WSMO mediator architecture.

3.1 Δ-Relations Definition

As a basis for efficient resource management by additional information on the teleological level as outlined above, a Δ-relation denotes the explicit logical relationship between functional resource descriptions. Following [2], this can most appropriately be described as the logical difference.

Hence, we refer to this a Δ-relation that consists of two elements: the **Δ-expression** states the logical difference between functional descriptions, and the **Δ-situation** that denotes the type of the relationship between them. Both can be computed for given functional descriptions. In order to provide a general definition that is adaptable to the semantics of functional descriptions in respective frameworks, we apply a set-theoretic model. Referring to [14] for details, the set-theoretic model defines that if ϕ is a functional description it is interpreted as a subset of the universe $\mathcal{U}$ (that is all possible instances of the ontologies used as terminologies in ϕ) that satisfies ϕ, i.e. $\phi \subset \mathcal{U}$.

Following this, the Δ-expression between two given arbitrary logical formulas ϕ and ψ is their union without their intersection: $\Delta_{\phi,\psi} = \phi \cup \psi \setminus (\psi \cap \phi)$. This means that Δ contains those elements that are models for either ϕ or ψ and not common to them. Considering the above example of two requests R_ϕ = 'finding restaurants in Innsbruck' and R_ψ = 'finding Italian restaurants in Innsbruck', Δ_{R_ϕ,R_ψ} would be 'all restaurants in Innsbruck that are not Italian'. Using de Morgan's laws, we can simplify this formula as follows: $\Delta_{\phi,\psi} = \phi \cup \psi \setminus (\psi \cap \phi) = \phi \setminus (\psi \cap \phi) \;\cup\; \phi \setminus (\psi \cap \phi) = \phi \backslash \psi \;\cup\; \psi \backslash \phi$. This states that the desired logical difference between ϕ and ψ is the union of ϕ without ψ and ψ without ϕ.

The Δ-situation denotes the type of relationship between formulas, respectively functional descriptions by commonly used keywords. In the example, R_ψ is a subset of R_ϕ; we denote is a subsumption relation between R_ϕ and R_ψ. As it holds in this situation that only those Web services usable for R_ϕ might be usable for R_ψ by no others, the information on the Δ-situation appears to be relevant with respect to the aspired usage for efficient resource management outlined above. Hence, we define five Δ-situations that naturally comply with the degrees of matching identified in [17], [14]. While in discovery these are used for denoting the type of commonality between logical expressions, we use them to denote the type of difference. The following defines the Δ-situations and the simplified computation of the corresponding Δ-expression:

1. **equal**: $\phi = \psi \Rightarrow \Delta = \emptyset$.
   this means that the models for ϕ and ψ are exactly the same so that there does not exists any logical difference between them.
2. **plugin**: $\phi \subset \psi \Rightarrow \Delta = \psi \backslash \phi$.
   this means that all models of ϕ are also models for ψ but not vice versa. We can also say that ϕ is subsumed by ψ.
3. **subsume**: $\phi \supset \psi \Rightarrow \Delta = \phi \backslash \psi$.
   as the opposite of the plugin situation, this means that all models of ψ are also models for ϕ but not vice versa. We say that ϕ is subsumes ψ. (the differentiation of the situations "subsume" and "plugin" gets important for enabling efficient reasoning mechanisms, as discussed below).

4. **intersecting**: $\phi \cap \psi \neq \emptyset => \Delta = \phi \backslash \psi \ \cup \ \psi \backslash \phi$.
   if there is no proper specialization or generalization but there exist models common for ϕ and ψ, then the Δ between them is their union without their intersection - i.e. we cannot simplify the computation of the Δ-expression.
5. **disjoint**: $\phi \cap \psi = \emptyset => \Delta = \emptyset$.
   if there does not exist any common model for ϕ and ψ, then we consider the Δ-expression to be empty as there is no correlation between the formulas.

The above definitions provide a general definition of Δ-relations between arbitrary logical formulas that can be applied for WSMO capabilities as follows. As the description of the functionality provided by a Web service as well as for the requested functionality in Goals, a WSMO capability is comprised of shared variables, preconditions and assumptions that denote the pre-state, and postconditions and effects that denote the post-state. While the four latter elements are defined by axioms, the scope of shared variables is the complete capability that allows specifying the coherence between the pre-state and post-state description elements of a capability. The intended semantics is that if the input provided is a valid model for the pre-state, then the execution of the Web service or the solution of the Goal will result in a post-state that is dependent of the respective pre-state. A formal semantics is under development at the point of writing, which is based on the notion of Abstract State Spaces wherein a Web service is understood as a set of state transitions from an initial to a termination state (see [7] for details).

Following this, we cannot write a WSMO capability definition in a single logical formula - at least not without respective signature renaming. Hence, we denote the Δ-relations between two WSMO capabilities $C_1 = (\phi_{pre}, \phi_{ass}, \phi_{post}, \phi_{eff})$ and $C_2 = (\psi_{pre}, \psi_{ass}, \psi_{post}, \psi_{eff}$ as a tuple of the Δ-relations between the corresponding description elements whereby the distinct Δ-relations are computed by the above methods. Hence, under consideration of all ontologies O and mediators M used in the description of two WSMO capabilities C_1, C_2, the logical relationship between them is defined as follows:

$$\mathcal{O}, \mathcal{M}, \mathcal{C}_1, \mathcal{C}_2 \ \models \ \Delta_{C_1,C_2} = (\Delta_{\phi_{pre},\psi_{pre}}, \Delta_{\phi_{ass},\psi_{ass}}, \Delta_{\phi_{post},\psi_{post}}, \Delta_{\phi_{eff},\psi_{eff}}) \quad (1)$$

This allows performing the desired reasoning tasks for improving the efficiency of resource management in Semantic Web service technologies. Thereby, we can compute the Δ-relations between the description elements of capabilities and reason on these. Dependent on what is to be achieved by working with delta-relations, we can also transform WSMO capability definitions into single logical formulas. Δ-relations between OWL-S Profile descriptions can be defined in a similar way.

The set-theoretic definition for computing the Δ-expression is transformed into the respective description language. For instance, when dealing with first-order logic expressions ϕ and ψ the Δ-expression between them is defined by $\Delta_{\phi,\psi} = (\phi \wedge \neg\psi) \vee (\neg\phi \wedge \psi)$; the definition is analogue for functional descriptions that use description logics or logic programming with respect their semantics.

3.2 Using $\varDelta$-Relations for Gaining Efficiency

After identifying the potential of teleological level mediation and definition of $\varDelta$-relations as the main mediation technique, the following exposes the benefits attainable for Semantic Web service technologies.

As outlined introductory, the main merit of teleological level mediation is to increase efficiency in Semantic Web service technologies. With respect to this, we distinguish two functional purposes for beneficially utilization: (1) support for *problem and functionality specification by reuse and refinement*, and (2) *creation of element ontologies with additional information on the teleological level.* While the former purpose mainly refers to support for creating the semantic description of goals and Web services, the latter facilitates efficiency in mechanisms for automated discovery and composition of Semantic Web services. We explain this in more detail.

For illustration purpose, let's consider the following example. A goal G_1 defines *buy product*, and another goal G_2 defines *buy ticket*, whereby *ticket* is sub-class of *product* in the used domain ontology. Considering the capability specification of G_1 to be ϕ, and the one of G_2 to be ψ, then there is a subsume situation $\phi \supset \psi$ so that $\varDelta = \phi \backslash \psi$. For the first usage scenario, imagine that G_1 already exists and some user wants to define G_2. As G_2 is a teleological refinement of G_1 (means: both goals have the same structure, but the object of interest in G_2 is narrower than the one of G_1), we can use a mediator $\mathcal{M}_{G_1,G_2}$ that contains $\varDelta_{G_1,G_2}$ for automatically deriving the specification of G_2 as it holds: $G_2 = G_1 \backslash \varDelta_{G_1,G_2}$. Similar, we can create the capability specifications of interrelated Web services. Following the concept of weakening and strengthening for describing Problem Solving Methods [10], this simplifies the creation of problem and functionality descriptions.

Besides, we attain additional information on the teleological relationship between elements that are interconnected by mediators with $\varDelta$-relations. Considering such an element collection as a graph, the goals and Web services represent the nodes and the mediators with $\varDelta$-relations denote the arcs that explicitly define the teleological relationship between the goals and Web services. We refer to such collections of semantically interlinked elements as *teleological element ontologies* that provide additional teleological level information and can be used for improving efficiency of central reasoning tasks for Semantic Web services.

For instance, referring to the above example, imagine that from previous runs of a Web Service discovery engine we have determined a set of Web Services that are applicable for resolving goal G_1: $WS_{G_1} = (WS_1, WS_2, ..., WS_n)$. Because of the situation $G_1 \supset G_2$ we know that the set of applicable Web Service for resolving G_2 can only be equal or a subset of those applicable for G_1: $WS_{G_2} \subseteq WS_{G_1}$. Hence, we can derive WS_{G_2} as those Web services out of WS_{G_1} that satisfy $\varDelta_{G_1,G_2}$ in the mediator $\mathcal{M}_{G_1,G_2}$ outlined above as it holds: $WS \in WS_{G_2} \leftarrow WS \in WS_{G_1} \wedge \; satisfy(WS, \varDelta_{G_1,G_2})$. Although becoming more complicated when taking $\varDelta$-relations between the goals and Web services into account (see Section 4), this shows that we can omit invocation of a discoverer for determining Web services satisfying G_2 - which most presumably is more expensive than checking this simple inference. Hence, teleological level mediation significantly decreases the reasoning effort in Semantic Web service technology by following the approach of gaining efficiency for automated problem solving by additional constraints between resource descriptions as presented in [9].

3.3 Integrating Teleological Level Mediation in WSMO

Completing teleological level mediation techniques, the following incorporates the outlined usage of Δ-relations into the WSMO mediators architecture in order to attain an integrated mediation model for Semantic Web services [20].

As shown in Figure 1, WSMO distinguishes four mediator types: *OO Mediators* that connect ontologies and provide data level mediation facilities. *GG, WG*, and *WW Mediators* connect goals and Web services. Each mediator connects source and target components denoted by the denomination prefix, and applies respective mediation techniques in order to resolve and handle the heterogeneities that can potentially arise between the source and target.

As mediation facilities, *GG, WG*, and *WW Mediators* can use OO Mediators for handling data level heterogeneities and may contain Δ-relations as the teleological level mediation definition. In addition, *WG Mediators* and *WW Mediators* can use a process mediator for resolving behavioral mismatches in communication or cooperation. We consider this mediation framework to be complete for Semantic Web services as it defines architectural components that apply appropriate mediation facilities for all heterogeneity types that can appear between the core elements of Semantic Web service systems.

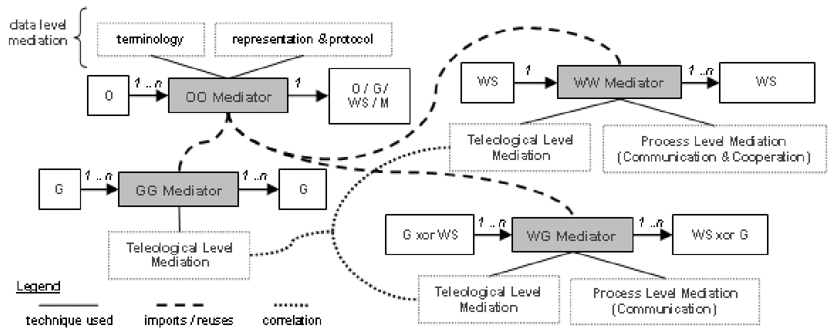

Fig. 1. WSMO Mediator Topology

There are two aspects to be mentioned for teleological level mediation in *GG Mediators, WG Mediators*, and *WW Mediators*. At first, we need to define the correlation between Δ and the source and target components of the mediator. As discussed above, WSMO capabilities are self-contained logical statements so that Δ-relations refer to complete capability. In consequence, we define the semantics of Δ-relations in WSMO mediators as follows:

- Δ defines the explicit logical relationship as the difference between the capabilities of the source S and the target T of a mediator $\mathcal{M}_{S,T}$ as defined in formula 1
- the Δ-situation is defined from the source component of a mediator to its target component (e.g. $plugin(S,T)$ denotes that $S \subset T$); as this information is needed for prosperous reasoning on teleological element ontologies, it is denoted in a non-functional type property of the Δ-relation definition
- in case of a proper generalization $plugin(S,T)$ or specialization $subsume(S,T)$, we can automatically attain $\mathcal{T}_{cap}$ from $\mathcal{S}_{cap}$ via Δ and vice versa:
  1. if $\mathcal{S}_{cap} \supset \mathcal{T}_{cap}$, then $\mathcal{T}_{cap} = \mathcal{S}_{cap} \backslash \Delta_{S,T}$ and $\mathcal{S}_{cap} = \mathcal{T}_{cap} \cup \Delta_{S,T}$
  2. if $\mathcal{S}_{cap} \subset \mathcal{T}_{cap}$, then $\mathcal{T}_{cap} = \mathcal{S}_{cap} \cup \Delta_{S,T}$ and $\mathcal{S}_{cap} = \mathcal{T}_{cap} \backslash \Delta_{S,T}$
  3. otherwise, T can not be attained directly from S via $\Delta_{S,T}$ or vice versa.

Secondly, there is a correlation between Δ-relations in *GG, WG Mediators* and *WW Mediators* denoted by the doted lines in Figure 1. In case the same goals and Web services are connected by respective mediators, we can derive new Δ-relations out of existing ones. For instance, referring to the introductory example of finding restaurants in Innsbruck, it holds that $\Delta_{R,WS_1} = \Delta_{R_O,R} \cup \Delta_{R_O,WS_1}$. Such correlations depend on the Δ-situations in the respective mediators and require further investigation that is out of the scope of this paper. However, this property allows learning mechanisms for Δ-relations for incrementally increasing the efficiency of Semantic Web service technology.

4 Evaluation by Example

This section demonstrates the usage and benefits of teleological level within exemplary scenarios in order to verify the above theoretical explorations. We first depict a simple scenario for improving the efficiency of Web service discovery along with exhibiting the modeling of Δ-relations in WSMO mediators, After that, we discuss more complex scenarios.

The following exemplifies the benefits of teleological level mediation for improving efficiency in Web service discovery as a core reasoning tasks within Semantic Web services. Discovery is concerned with detecting appropriate Web services for a given request or application scenario [14]. Therefore, we re-consider the introductory example of finding restaurant information in Innsbruck with the following goals and Web services involved:

1. G_O: a goal for finding a restaurant in Innsbruck
2. G_1: a goal for finding an Italian restaurants in Innsbruck
3. WS_1: a Web service 'Innsbruck Hotel and Restaurant Guide' that provides information on all hotels and restaurants in Innsbruck
4. WS_2: a Web service 'Tyrol Restaurant Guide' that covers restaurants in Tyrol (the state of Austria where Innsbruck is located in)
5. WS_3: a Web service 'Traditional Cuisine in Innsbruck' that provides information on Tyrolean restaurants in Innsbruck

Obviously, these goals and Web services are related to each other. For the goals, G_1 appears to be a specialization of G_0, and the Web services seem to be applicable for resolving these goals. If we would have no additional teleological information, we would have to create each goal separately and run a complete, most likely complex discovery process for determining which Web services can be used for resolving the goals. We assume the following situation for demonstrating how the efficiency improvement by teleological mediation: G_0 already exists, and we know from a previous discovery run that all Web services WS_1, WS_2, WS_3 can be used for resolving G_0. Now, a user wants to create G_1 and find usable Web services. Therefore, we define a teleological element ontology that consists of the goals, the Web services, GG and WG Mediators. Figure 2 shows this, including the Δ-situation and the direction of the mediators.

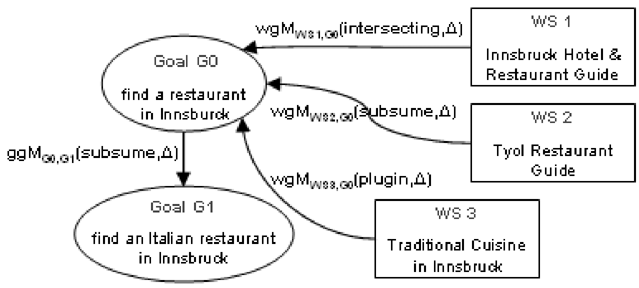

Fig. 2. Teleological Element Ontology Example

The following specifies the goals, Web services, and mediators as WSMO elements using the Web Service Modeling Language WSML [5], and exemplifies the modeling of Δ-relations. The listing below shows the ontology used as the terminology in this example, the goals G_0 and G_1, and the GG Mediator ggM_{G_0,G_1} (all elements are intended for academic demonstration purpose only and hence are very limited).

```
// Ontology used as terminology in example
namespace { _"http://www.wsmo.org/ontologies/mediate05example#",
  dc _"http:// purl .org/dc/elements/1.1#",
  loc _"http:://www.wsmo.org/ontologies/location#" }
ontology _"http://www.wsmo.org/ontologies/mediate05example"
 importsOntology _"http://www.wsmo.org/ontologies/location#"
 // an ontology for locations and addresses
 concept restaurant
     name ofType _string
     type ofType restaurantType
     address ofType loc#address
 // pre−defined instances
  austria memberOf loc#country
  tyrol memberOf loc#state
     name hasValue "Tyrol"
     inCountry hasValue austria
 innsbruck memberOf loc#city
```

```
      name hasValue "Innsbruck"
      inState hasValue tyrol

// Goal definitions (only postconditions modeled here)
namespace { _"http://www.wsmo.org/mediate05/G0#",
  o _"http://www.wsmo.org/ontologies/mediate05example#" }
goal _"http://www.wsmo.org/mediate05/G0"
// G0 — find a restaurant in Innsbruck
 importsOntology _"http://www.wsmo.org/mediate05/G0#"
  capability
   postcondition
      definedBy
        ?x[address.inCity hasValue = innsbruck] memberOf o#restaurant .

namespace { _"http://www.wsmo.org/mediate05/G1#",
  o _"http://www.wsmo.org/ontologies/mediate05example#" }
goal _"http://www.wsmo.org/mediate05/G1"
// G1 — find Italian restaurants in Innsbruck
 importsOntology _"http://www.wsmo.org/mediate05/G0#"
  capability
   postcondition
      definedBy
        ?x[type hasValue italian ,
            address.inCity hasValue innsbruck]
        memberOf o#restaurant .

// GG Mediator between G0 and G1
namespace { _"http://www.wsmo.org/mediate05/ggm#",
  o _"http://www.wsmo.org/ontologies/mediate05example#" }
ggMediator _"http://www.wsmo.org/mediate05/ggm"
 importsOntology _"http://www.wsmo.org/mediate05/G0#"
 source _"http://www.wsmo.org/mediate05/G0"
 target _"http://www.wsmo.org/mediate05/G1"
 Δ−relation
   nonFunctionalProperties
     dc#type hasValue subsume
   endNonFunctionalProperties
   definedBy
        ?x [type hasValue ?type,
        address.city hasValue innsbruck
        ] memberOf o#restaurant and
        not(?type = italian ).
```

The GG Mediator ggM_{G_0,G_1} has the source goal G_0 and the target goal G_1. The difference between them is that G_0 defines all restaurants as the desired information, while G_1 only desires Italian restaurants in Innsbruck. Hence, as $G_0 \supset G_1$, the Δ-relation is a subsumption from the source to the target goal.

The following shows the capabilities of the three Web services and the Δ-relations of the respective WG Mediators with a Web service as the source and G_0 as the target component. Each WG Mediator carries a Δ-relation that denotes the explicit difference between the capabilities of the respective Web service and G_O. The Δ-situation as defined in Figure 2 become obvious when considering the capabilities of G_O and the respective Web services. We here omit structural WSML definitions due to length limitations and only model Web service capability postconditions as this is sufficient for demonstration purpose.

```
// WS1 'Innsbruck Hotel and Restaurant Guide' capability postcondition
  capability
    postcondition
      ?x[address.inCity hasValue innsbruck] and
      (?x memberOf o#restaurant or ?x memberOf o#hotel).
```

```
// WS2 'Tyrol Restaurant Guide' capability postcondition
 capability
  postcondition
   ?x[address.inState hasValue tyrol] memberOf o#restaurant.

// WS3 'Traditional Cuisine in Innsbruck' capability postcondition
 capability
  postcondition
   ?x[class hasValue traditional, address.inCity hasValue innsbruck] memberOf o#restaurant.

// WG Mediator between WS1 and G0
wgMediator _"http://www.wsmo.org/mediate05/wgm1"
 source _"http://www.wsmo.org/mediate05/WS1"
 target _"http://www.wsmo.org/mediate05/G0"
 Δ−relation
  nonFunctionalProperties
   dc#type hasValue intersecting
  endNonFunctionalProperties
  definedBy
   ?x[address.inCity hasValue innsbruck] memberOf o#hotel.

// WG Mediator between WS2 and G0
wgMediator _"http://www.wsmo.org/mediate05/wgm2"
 source _"http://www.wsmo.org/mediate05/WS2"
 target _"http://www.wsmo.org/mediate05/G0"
 Δ−relation
  nonFunctionalProperties
   dc#type hasValue subsume
  endNonFunctionalProperties
  definedBy
   ?x[address.inCity hasValue ?city] memberOf o#restaurant and not(?city = innsbruck).

// WG Mediator between WS3 and G0
wgMediator _"http://www.wsmo.org/mediate05/wgm3"
 source _"http://www.wsmo.org/mediate05/WS3"
 target _"http://www.wsmo.org/mediate05/G0"
 Δ−relation
  nonFunctionalProperties
   dc#type hasValue plugin
  endNonFunctionalProperties
  definedBy
   ?x[type hasValue ?type] memberOf o#restaurant and not(?type = traditional).
```

Now we can discuss how the additional teleological mediation information can be beneficially utilized for gaining efficiency in reasoning tasks for Semantic Web services. As a major one, discovery is concerned with determining appropriate Web services for resolving a given goal. Therefore, semantic techniques are applied that determine logical relationship between functional service and goal descriptions in order to increase the accuracy of discovery results. As several aspects like valid pre-state and post-state detection need to be taken into account, adequate discovery engines for Semantic Web services consists of complex reasoning procedures (see [17], [13], [15], [24]).

For explaining how the need for such expensive discovery procedures can be omitted, we assume that all elements of the teleological element ontology shown in Figure 2 are given (i.e. all goals, Web services, and mediators). Also, we assume to know from a previous discovery run that WS_1, WS_2, WS_3 are usable for resolving G_O; it holds that only these or a subset can be usable for resolving G_1 because the subsume situation between the capabilities of G_O and G_1. For a Web service to be usable for the G_1 it has to satisfy its object definition that is strengthened, i.e narrowed in comparison to the one of G_O. This can be determined via the Δ defined in the GG Mediator ggM_{G_O,G_1}. If a Web service that is in the discovery result of G_O satisfies Δ_{G_O,G_1}, then it is usable for resolving G_1 so that: $G_O \supset G_1 \wedge (usable(WS, G_1) \leftarrow usable(WS, G_O) \wedge satisfied(WS, \Delta_{G_O,G_1}))$. Evaluating this rule determines WS_1, WS_2 to be usable for resolving G_1 while WS_3 does not satisfy the Δ_{G_O,G_1}.

This example has discussed the simplest setting of efficient Web service discovery on basis of Δ-relations. However, the procedure can get more complex in case that a different Δ-relation exists between the source and target goal in a GG Mediator. While there can not exist any Web service that is usable for resolving G_y but not for G_x if the Δ-situation is $subsume(G_x, G_y)$, this can be the case for different Δ-situations. In such cases, the relationship between the Δ-relations in GG Mediators and those in WG Mediators needs to be taken into consideration. If there is a concatenation of subsumption Δ-relations between goals and Web services we do not even need to evaluate the Δ-relations as $subsume(WS, G_y) \leftarrow subsume(G_x, G_y) \wedge subsume(WS, G_x)$; on the other hand, in case of a concatenation of intersecting Δ-situations we possibly need to use a discoverer as the teleological mediation information are not sufficient for ensuring correctness of the discovery results. Hence, beneficial usage of Δ-relations for efficient discovery require more complex algorithms with respect to all possible combinations of Δ-situations that can occur in GG and WG Mediators. We do not discuss this any further as it exceeds the aim and scope of this paper.

We have demonstrated efficiency improvement for discovery as one main reasoning tasks for Semantic Web services. However, we can follow the same approach for improving efficiency within other mechanisms that are concerned with teleological level information like service composition. Therefore, we can define GG Mediators that establish a collection of sub-goals $G_{sub1}(G_x), G_{sub2}(G_x), ...$ for some complex goal G_x in the sense of a functional decomposition. If we do not discover any Web service that is capable of resolving G_x but some services for its sub-goals, we have determined the input required by a Web service composition engine for dynamically constructing a suitable execution model of the services usable for the sub-goals. Such application scenarios of teleological mediation need to deal with more complex relationships of Δ-relations that we consider to be future work.

5 Related Work

We are not aware of any other approach that identifies the need for mediation on the teleological level for Semantic Web services or provides support for this. Nevertheless, the following outlines work that has inspired the approach presented in this paper.

The need for efficient resource management has been revealed throughout our work on reasoning mechanisms for Semantic Web services with respect large-scale applicability and the performance problem of complex reasoning systems. Existing approaches like [11], [24] address this by defining classifications or architectural constraints as the basis for layered architectures that subsequently narrow the search-space, i.e. reducing the number of elements that needs to be inspected in complex reasoning mechanisms. However, these techniques do not explicitly express the teleological relationship between resources and thus do not adequately support reasoning on additional teleological information in a way comparable to the one we have presented.

Our approach for teleological mediation has been inspired by the concept of refinement in the UPML framework for describing Problem Solving methods [10]. Therein, so-called *Refiners* define additional constraints referred to as Δ that bridge the teleological gap between goals or tasks to be achieved, the problem solving method that specifies the reasoning process, and the domain knowledge used for achieving the task [2]. This significantly decreases the required coverage for functionally describing a problem solving method as several aspects can are eliminated by Δs, hence allows gaining efficiency in the reasoning process for automatically resolving a goal [9]. This work has served as a basis for our definition of Δ-relations and their usage for improving efficiency in reasoning mechanisms for Semantic Web Services.

The idea of using information on the difference between resources for increasing efficiency in handling them is also applied in other technologies. For instance, video compression techniques like MPEG use so-called *delta frames* that only specify the changes between consecutive pictures; these are significantly smaller with respect to the amount of data required for specification and hence reduce the file size of videos [3]. This coincides with the approach of additional teleological information for reducing the reasoning effort for Semantic Web service techniques as we have presented here.

6 Conclusions and Future Work

This paper has introduced the teleological level as a novel aspect of mediation for Semantic Web services. This level deals with heterogeneities that arise between functional descriptions of Web services and related elements. We have defined Δ-relations that explicitly denote the teleological difference between functional element descriptions, integrated them into the mediation framework of the Web Service Modeling Ontology WSMO, and outlined how these additional information can be beneficially utilized for improving the efficiency of reasoning mechanisms for Semantic Web services.

Teleological level mediation as presented here is different from data and process level mediation. While the latter are concerned with techniques for establishing interoperability if this is not given a priori by resolving mismatches, teleological level mediation is concerned with improving the efficiency of Semantic Web service technologies. The elements that are connected via mediators in a teleological element ontology can reside in a functional manner without the additional teleological information. In the example on efficiency improvement for discovery we can accomplish the same correct discovery result by invoking a discovery engine instead of evaluating the Δ-relations. However, efficiency of core technologies for handling Semantic Web services is a crucial issue with respect to large-scale, industrial strength applicability. As teleological mediation with Δ-relations can significantly improve efficiency, we consider this to be a beneficial mediation technique for Semantic Web services.

While this paper presents the foundation of teleological level mediation, future efforts will be concerned with integrating this technique into functional components for discovery and composition of Semantic Web services as well as elaboration of advanced algorithms for enhanced reasoning on teleological element ontologies. In a longer term, we will also consider techniques for automatically learning Δ-relations within Semantic Web service environments that enable dynamic improvement of a system's efficiency during its life time.

Acknowledgements

This material is based upon work funded by the EU under the DIP project (FP6 - 507483) and by the Science Foundation Ireland under Grant No. SFI/02/CE1/I131. The authors would like to thank the members of the WSMO working group (www.wsmo.org) and dedicate special thanks to Uwe Keller for fruitful advice and input to the presented work.

References

1. V. Alexiev, M. Breu, J. de Bruijn, D. Fensel, R. Lara, and H. Lausen. *Information Integration with Ontologies.* Wiley, West Sussex, UK, 2005.
2. V. R. Benjamin, D. Fensel, and R Straatman. Assumptions of Problem-Solving Methods and their Role in Knowledge Engineering. In *Proceedings of the European Conference on Artifical Inteliigence (ECAI 1996), Budapest, Hungary*, 1996.
3. A. C. Bovik (ed.). *Handbook of Image and Video Processing.* Academic Press, 2000.
4. E. Cimpian and A. Mocan. WSMX Process Mediation Based on Choreographies. In *Proceedings of the 1st International Workshop on Web Service Choreography and Orchestration for Business Process Management at the BPM 2005, Nancy, France*, 2005.
5. J. de Bruijn (ed.). The Web Service Modeling Language WSML. WSML Deliverable D16.1 final version 0.2, 2005. available from http://www.wsmo.org/TR/d16/d16.1/v0.2/.
6. J. B. Domingue, D. Roman, and M. Stollberg (eds.). Web Service Modeling Ontology (WSMO) - An Ontology for Semantic Web Services. Position Paper at the W3C Workshop on Frameworks for Semantics in Web Services, June 9-10, 2005, Innsbruck, Austria, 2005.
7. Lausen (ed.). Functional Description of Web Services. WSML Deliverable D28.1, 2005. Most recent version available at: http://www.wsmo.org/TR/d28/d28.1/.
8. D. Fensel and C. Bussler. The Web Service Modeling Framework WSMF. *Electronic Commerce Research and Applications*, 1(2), 2002.
9. D. Fensel and R. Straatman. The Essence of Problem-Solving Methods: Making Assumptions to Gain Efficiency. *International Journal of Human-Computer Studies*, 48(2):181–215, 1998.

10. D. Fensel et al. The Unified Problem Solving Method Development Language UPML. *Knowledge and Information Systems Journal (KAIS)*, 5(1), 2003.
11. F. Giunchiglia, M. Yatskevich, and E. Giunchiglia. Efficient Semantic Matching. In *Proceedings of the 2nd European Semantic Web Conference (ESWC 2005), Crete, Greece*, 2005.
12. A. Haller, E. Cimpian, A. Mocan, E. Oren, and C. Bussler. WSMX - A Semantic Service-Oriented Architecture. In *Proceedings of the International Conference on Web Service (ICWS 2005), Orlando, Florida*, 2005.
13. U. Keller, R. Lara, H. Lausen, A. Polleres, and D. Fensel. Automatic Location of Services. In *Proceedings of the 2nd European Semantic Web Conference (ESWC 2005), Crete, Greece*, 2005.
14. U. Keller, R. Lara, and A. Polleres (eds.). WSMO Web Service Discovery. Deliverable D5.1, 2004. available at: http://www.wsmo.org/TR/d5/d5.1/.
15. M. Kifer, R. Lara, A. Polleres, C. Zhao, U. Keller, H. Lausen, and D. Fensel. A Logical Framework for Web Service Discovery. In *Proc. of the ISWC 2004 workshop on Semantic Web Services: Preparing to Meet the World of Business Applications, Hiroshima, Japan, Nov. 2004*, 2004.
16. H. Lausen, A. Polleres, and D. Roman (eds.). Web Service Modeling Ontology (WSMO). W3C Member Submission 3 June 2005, 2005. online: http://www.w3.org/Submission/WSMO/.
17. L. Li and I. Horrocks. A software framework for matchmaking based on semantic web technology. In *Proceedings of the 12th International Conference on the World Wide Web, Budapest, Hungary*, 2003.
18. D. Martin (ed.). OWL-S: Semantic Markup for Web Services. W3C Member Submission 22 November 2004, 2004. online: http://www.w3.org/Submission/OWL-S.
19. S. McIlraith, T. Cao Son, and H. Zeng. Semantic Web Services. *IEEE Intelligent Systems, Special Issue on the Semantic Web*, 16(2):46–53, 2001.
20. A. Mocan, E. Cimpian, and M. Stollberg (eds.). WSMO Mediators. Deliverable D29, 2005. Most recent version available at: http://www.wsmo.org/TR/d29/.
21. A. Mocan (ed.). WSMX Data Mediation. WSMX Working Draft D13.3, 2005. available at: http://www.wsmo.org/TR/d13/d13.3/v0.2/.
22. M. Moran and A. Mocan. Towards Translating between XML and WSML based on mappings between XML Schema and an equivalent WSMO Ontology. In *Proceedings of the WIW 2005 Workshop on WSMO Implementations, Innsbruck, Austria*, 2005.
23. M. Paolucci, N. Srinivasan, and K. Sycara. Expressing WSMO Mediators in OWL-S. In *Proceedings of the workshop on Semantic Web Services: Preparing to Meet the World of Business Applications held at the 3rd International Semantic Web Conference (ISWC 2004) , Hiroshima, Japan*, 2004.
24. M. Stollberg, U. Keller, and D. Fensel. Partner and Service Discovery for Collaboration Establishment on the Semantic Web. In *Proceedings of the Third International Conference on Web Services, Orlando, Florida*, 2005.
25. G. Wiederhold. Mediators in the architecture of the future information systems. *Computer*, 25(3):38–49, 1994.