# Semantic System Integration – Incorporating Rule-based Semantic Bridges into BPEL Processes

Nils Barnickel, Ralf Weinand, Matthias Fluegge
Fraunhofer Institute for Open Communication Systems (FOKUS),
Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
{Nils.Barnickel, Ralf.Weinand, Matthias.Fluegge}@fokus.fraunhofer.de

**Abstract.** This paper describes how semantic bridges realized in terms of rule-based ontology mappings can be incorporated into BPEL processes. The approach is explained by applying it to a semantic system integration scenario in the eBusiness domain defined as the "purchase order mediation" scenario in the context of the Semantic Web Service Challenge[1]. The presented approach relies strongly on the existing Web standards and is based on widely adopted open source software components.

**Keywords:** semantic interoperability, semantic mediation, semantic bridge, ontology mapping, SWRL, BPEL, SWS-Challenge

## 1 Introduction

The advent of the service-oriented architecture (SOA) model and its implementation in the form of Web services has contributed significantly to facilitate the technical integration of information systems. However, semantic interoperability, i.e. the semantically sound exchange of data between heterogeneous systems, still represents a challenging task. The main reason is that domain- and application-specific requirements have produced and will always produce different information models for one and the same problem.

The alignment of heterogeneous information models is impeded by the fact that they are often represented in a semantically poor manner, focusing only on structural specifications for data exchange. This makes system integration mainly a manual effort and it requires considerable technical skills to define appropriate syntactical transformations. The application of Semantic Web technologies to system integration problems promises to mitigate this problem since the formal definition of semantics paves the way for (semi-)automated data and process mediation. Moreover, shifting information integration from the structural to the conceptual level represents a further step towards the ultimate goal of SOA, namely to align business and IT.

[1] http://sws-challenge.org/wiki/index.php/Scenario:_Purchase_Order_Mediation

## 1.1 Challenges and basic approach

A light-weight and effective semantic system integration approach is the use of semantic bridges as described in [1]. The basic idea is to wrap existing information resources with semantically described Web services and to leverage rule-based ontology mappings in order to achieve interoperability and to reduce manual efforts in the composition and execution of heterogeneous Web services. The paper at hand will briefly present how this approach can be applied to implement the "purchase order mediation" scenario as defined in the context of the Semantic Web Service Challenge[2] (SWSC).

The interacting systems of the SWSC mediation scenario mainly differ with regard to the following aspects

1. *data formats* (i.e. granularity and denotation of data elements) and
2. *interaction patterns* (order and granularity of operations)

The approach presented in this paper will address the first issue by applying semantic bridges to mediate between different information models and representations. The second issue will be addressed by using the Business Process Execution Language (BPEL) and an appropriate BPEL-compliant execution engine to orchestrate the services provided by both parties. This paper does not cover goal-oriented plan creation (compared to other approaches such as WSMO [2] or SWSF [3]) and leaves the planning task (i.e. which services to include at which part into the composition) to the business domain process expert (cf. section 1.2 Related Work). Thus, this paper presents a lightweight approach to reduce manual semantic mediation efforts by integrating semantic bridges into BPEL.

Semantic bridges describe the relations between entities in business information models that are defined in different ontologies but have a similar meaning. At the same time semantic bridges define appropriate mappings in order to translate instances of such entities or so called concepts. Ideally such a mapping can be included directly and transparently in the reasoning processes, which allows for drawing conclusions and thus provides the foundation for tool-supported semi-automatic semantic mediation.

The core concept of semantic bridges lies in the shift of semantic mediation from the structural to the conceptual abstraction level in order to reduce efforts for achieving semantic interoperability. Moreover, semantic bridges cannot just be applied in the execution phase but also in the design phase of a business process. A matching engine transparently applies semantic bridges and performs the reasoning over semantically described relationships (such as inheritance or equality between concepts), thus enabling a composition tool to semi-automatically support the design of interaction patterns by issuing recommendations for suitable assignments between output and input parameters of different Web services. Consequently, achieving semantic interoperability requires less manual integration efforts.

A promising approach to meet the described requirements is the use of expressive rule languages to capture mappings and to enable the direct application of the specified mappings for corresponding instance transformations. Logic-based rules are computationally complete. Hence, by defining semantic bridges in terms of logic-

[2] http://sws-challenge.org/wiki/index.php/Scenario:_Purchase_Order_Mediation

based rules, any kind of mapping relation (one-to-one, one-to-many, many-to-many) can be described. The absence of technical transformation code increases ease and maintainability of the semantic bridges. Furthermore and most importantly, an inference service can directly apply the rules as part of its reasoning process, i.e. the transformation of concept instances and their correct classification as well as potential further conclusions are handled in a well-integrated manner. When applying the approach in combination with existing ontology mapping tools [4,5] which allow to semi-automatically define the mappings as semantic bridge rules, manual integration efforts can be reduced substantially.

It has been recognized that the success of Semantic Web technologies relies on the reuse and integration of existing Web standards. The most widely-used standard for the composition of Web services is BPEL. A considerable number of mature BPEL-compliant process execution engines testify the broad industrial support for this standard which provides a rich set of control and data flow constructs for defining and aligning the interactions between the participating actors in a business process. The solution outlined in this paper raises the claim of being not only of theoretical but also of practical relevance. Consequently, the approach described in [1] was extended towards semi-automated data mediation in Web service processes that are formalized in terms of the BPEL standard.

The main challenge on this regard is to find a suitable mapping between different abstraction levels: While at design time ontologies and rules are used for data representation, data flow and mediation, BPEL execution engines make use of hierarchically structured XML Schema types, XPath and XSLT transformations. In order to face this challenge, the starting point is to exploit the RDF/XML serialization of ontologies for data representation on the BPEL level. Furthermore, BPEL enhancements have to be developed to integrate semantic bridges and to support data flow specifications in terms of rules. These enhancements are implemented as external functions that can be plugged into BPEL engines using a standardized extension mechanism as described in more detail in section 2. Also the application of rule-based semantic bridges and their integration into the data flow will be illustrated in section 2 for the "purchase order mediation" scenario.

### 1.2 Related Work

The long term vision behind Semantic Web services is to enable dynamic goal-oriented service composition and to use powerful inference engines and matchmaking mechanisms in order to automate the whole composition process including discovery, composition, execution and interoperation of Web services. Research background comes from the Semantic Web community on the one hand and from the field of dynamic planning in artificial intelligence research on the other hand. Significant work has been done in the context of WSMO [2]. WSMO includes a mediator concept to deal with the interoperation problems between Semantic Web services. The approach considers specific mediator services which perform translations between ontologies. These mediators attempt to reconcile the differences between goals of Web services. In the context of the SWS-Challenge this approach has been applied in [6]. The approach presented in this paper does not cover goal-oriented plan creation as it intentionally leaves the planning task (i.e. which services to include at which part into the composition) to the business domain process expert. Consequently, as presented in section 1.1, a more lightweight approach to semantic mediation has been developed.

The core concept of this approach is the integration of ontology mappings into BPEL processes. The ontology mappings are described through a set of description logic-based bridging axioms which refer to concepts or properties of a source ontology and specify how to express them in a target ontology. Thus, bridging axioms can be realized as rules which describe the transformation. The advantage of this rule-based approach is that reasoning over the described mappings can be applied straight forward as the transformation rules can be integrated into the regular ontology inference process, e.g. classifying etc. In particular, the presented approach in this paper applies SWRL rules, which are broadly supported by available Semantic Web frameworks and it makes use of the facet analysis classification mechanism (cf. section 2 Scenario Realization) supported by standard OWL semantics.

## 2 Scenario Realization

In order to be able to apply our mediation concept to the "purchase order mediation" scenario provided by the SWSC workshop we have to introduce several conceptual components which are numbered from one to five as illustrated in Fig. 1.

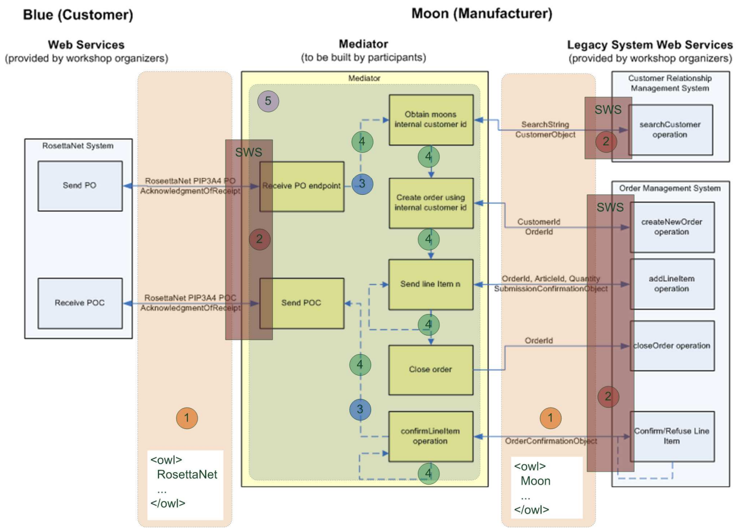


**Fig. 1.** Purchase order mediation scenario with highlighted conceptual components for semantic mediation (1) – (5).

In the following section the conceptual components for semantic mediation and their realization based on widely adopted open source products is described in detail.

### 2.1 Heterogeneous Domain Ontologies

For our scenario solution we assume that two different ontologies have been developed by RosettaNet domain experts and independently by domain experts of the Moon legacy system. The approach of multiple independent ontologies on the domain level takes into account the realistic point of view that existence of a single global ontology can not be assumed to cover all autonomous actors and organizations across various domains in a system integration scenario. Fig. 2 shows an outline of these heterogeneous information models which are formalized in the OWL [7] ontology language using defined classes. The information models differ in their semantic sub-graph. As the concept *Partner* in the RosettaNet ontology is defined in terms of three object properties a semantically corresponding concept *Customer* in the Moon ontology just features two object properties containing the same information, however defined at a lower level at granularity. By modeling these concepts as defined classes, corresponding OWL individuals can be easily classified by a reasoner supporting OWL facet analysis.

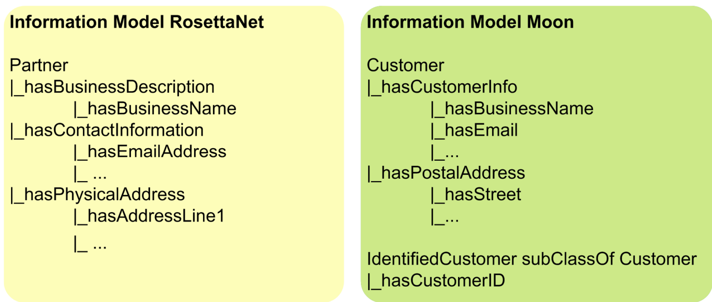


**Fig. 2.** Heterogeneous ontologies representing the different information models of the Blue system (RosettaNet) and the Moon system (legacy).

The existing XSD-based messages such as the PIP3A4 PO request message that is sent to the mediator by the Blue system are lifted to the concepts described in these heterogeneous domain ontologies using Semantic Web services, which are described in the following.

### 2.2 Semantic Web Services

The Web services provided in the scenario are annotated with concepts from the domain ontologies described above by applying the OWL-S ontology [8] for realizing Semantic Web services. Input and output parameters of Web services are linked to ontology concepts. For instance, in the scenario the Semantic Web service for the Moon CRM is expecting the defined class *Customer* which – among others – defines the property *hasBusinessName*. This property is used as the search criteria for the customer lookup. If a customer with the given name is found, an *IdentifiedCustomer* OWL individual is returned containing all customer attributes supplied by the CRM system. Furthermore, lifting and lowering definitions for converting the incoming and outgoing XSD instances to OWL instances are defined. This so called grounding mechanism in OWL-S is based on XSL transformations which are supposed to be developed by domain experts who semantically enrich their existing portfolio of Web services. OWL-S provides the main advantage in terms of tool support compared to other light weight approaches focusing mainly on service input and output annotations. In this regard the otherwise well fitting candidate SAWSDL [9] has not been chosen as a Java-API that fully executes Semantic Web services is not yet available. Therefore, the Mindswap OWL-S API has been applied to get programmatic access to read, execute and write OWL-S service descriptions. However, XSLT does not work on the same data model as OWL and thus can only operate on the OWL serialization. In our context the RDF/XML-ABBREV serialization applied in the OWL-S API implementation does not allow to exploit the full potential of polymorphism. When a polymorph individual is serialized using the RDF/XML-ABBREV format one of the types it holds is non-deterministically selected and the last fragment of the types URI is taken for the opening tag for this individual's XML serialization. The other types are expressed by separate *<rdf:type.../>* sub elements. This varying structure complicates the development of XSLT code dramatically. To overcome this weakness, the OWL-S API, has been adjusted accordingly. Now internally the basic RDF/XML serialization is applied. This means that all types are represented equally as sub elements, which allows to define straighter XSL transformations. Hence, the mapping from polymorph OWL serializations to single typed XML Schema instances can be achieved in terms of XSLT rules that match exactly the type which has been defined in the OWL-S input description.

The processing of the Web service results is less complicated as the received message parts correspond to the data model XSLT was designed for. XSLT rules can easily match the XML Schema instances and fill predefined skeletons of serializations of OWL individuals. It has to be mentioned that further modifications of the OWL-S API were necessary in order to provide support for WSDL messages typed via XML-elements and for operation calls based on Document/Literal communication as used in the provided Web services of the scenario.

### 2.3 Semantic Bridges

Obviously the *Partner* and *Customer* concepts presented above cannot be exchanged between communicating partners by default, although they represent the same conceptual idea. In order to mediate between these concepts the domain experts of both domains have to define mappings – so called semantic bridges – between the two heterogeneous ontologies. These semantic bridges are formalized as rules based on the Semantic Web Rule Language (SWRL) [10] and are made publicly available.

The application of rule-based semantic bridges for semantic mediation can be illustrated based on the simple example of two ontology concepts (*Partner* and *Customer*) which, although representing intuitively the same concept, have been defined independently in separate ontologies (cp. Fig. 2) and hence differ in their semantic sub-graph.

By applying the semantic bridge rules, an instance of type *Partner* is furnished with additional properties e.g. with *hasCustomerInfo* combining the values of the *BusinessDescription* and the *ContactInformation* properties *hasBusinessName* and *hasEmailaddress* as illustrated in Fig. 3.

Having the class definitions on hand, a reasoner is now able to classify the instance as a member of the defined class Customer, since all required properties (including *hasCustomerInfo*) are present. Thus, within the scope of the mediation process any service, independently to which domain it belongs, can now make use of this instance as it is polymorph of type *Partner* and *Customer*, i.e. semantic interoperability has been established.

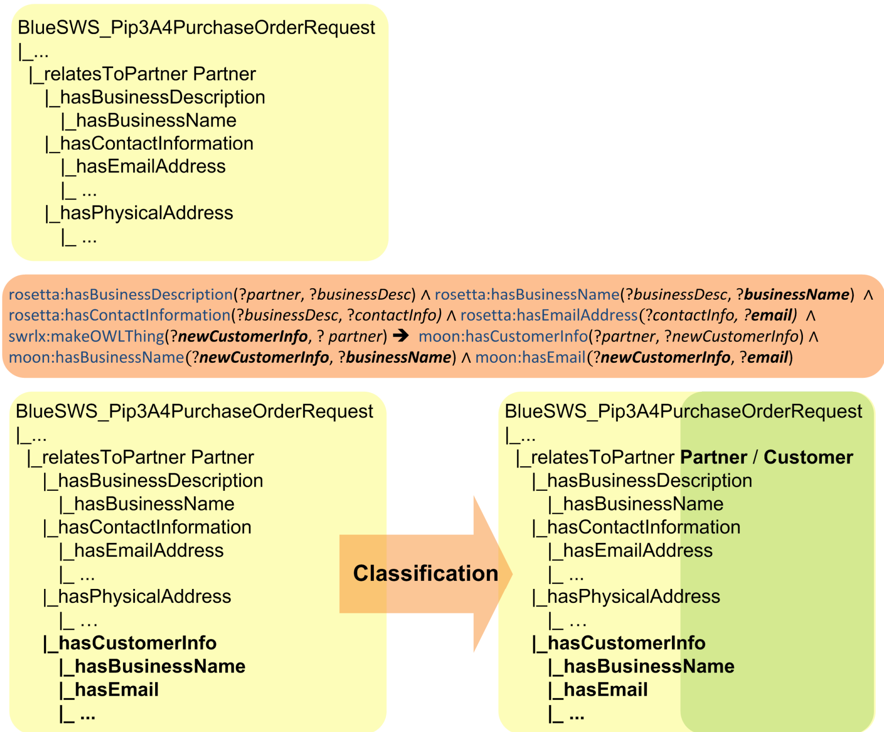


**Fig. 3.** Semantic Bridge and polymorph classification example preserving object identity.

Consequently, the transformed and reclassified individual can now be used as input for querying the Moon CRM. Thus, by applying the semantic bridge, the heterogeneous data formats of the Blue system and the Moon system have been successfully mediated. Using traditional XML-based Web services without semantic annotations, such an approach would not be feasible. As it has been argued in [11], the static type bindings do not allow for effective polymorphism; in particular XML Schema lacks inheritance relations to be exploitable for matching.

Technically, a semantic bridge is integrated into the BPEL process by using a standardized extension mechanism in terms of a custom XPath function call within an

Assign-Copy activity. The custom XPath function is implemented as a Java component based on the Jaxen framework and thus can be integrated in any BPEL execution engine that supports the configuration of custom XPath functions. Internally, the Java component that executes the semantic bridge relies on the Protégé-OWL API, which depends on the OWLAPI, Jena and Pellet for handling ontologies and performing DL-reasoning and on the Jess-rules engine which executes the SWRL rules [12].

### 2.4 Rule-based Data Flow

The usage of SWRL rules to mediate between heterogeneous ontology concepts is similarly applied in order to express the dataflow within the BPEL process. We distinguish between rules defining a semantic bridge and rules defining data flow within the process. Semantic bridges are developed on an ontology level independently of actual application scenarios. Hence, they can be reused for various integration scenarios between the involved parties. Rules defining the data flow are included into the BPEL process at design time of the integration scenario. In our scenario realization the data flow rules have been defined manually. However, as described in section 3 and in [1] the approach of using description logic-based Semantic Web services provides the foundation to generate the data flow in a semi-automatic manner. The concept of rule-based data flow is illustrated in Fig. 4.

The illustrated example shows the rule-based data flow integrated into the BPEL process in order to realize the hidden assumption of the purchase order mediation scenario: Although the search for a customer in the CRM system provides the mediator already with information such as *AddressInfo* or *ContactInfo*, still the information that is provided in RosettaNet messages should be used instead.

The first rule illustrated in Fig 4. binds the polymorph *Partner/Customer* individual holding the RosettaNet-based *PhysicalAddress* and *ContactInformation*, etc. Subsequently, the rule attaches the individual to a newly generated *NewOrderRequest* individual that is created in terms of SWRL built-Ins generated *NewOrderRequest* individual, The *NewOrderRequest* individual acts as the umbrella container for the input parameters of the Moon *CreateNewOrder* Semantic Web service.

The second rule attaches the *hasCustomerID* property received from the Moon *CustomerLookupResponse* Semantic Web service to the afore assigned *Customer* individual. Taking into account the corresponding OWL class definition, the individual is then classified as an *IdentifiedCustomer*. Thus, the above described assumption is anticipated and the *IdentifiedCustomer* individual can be used properly in the further process flow.

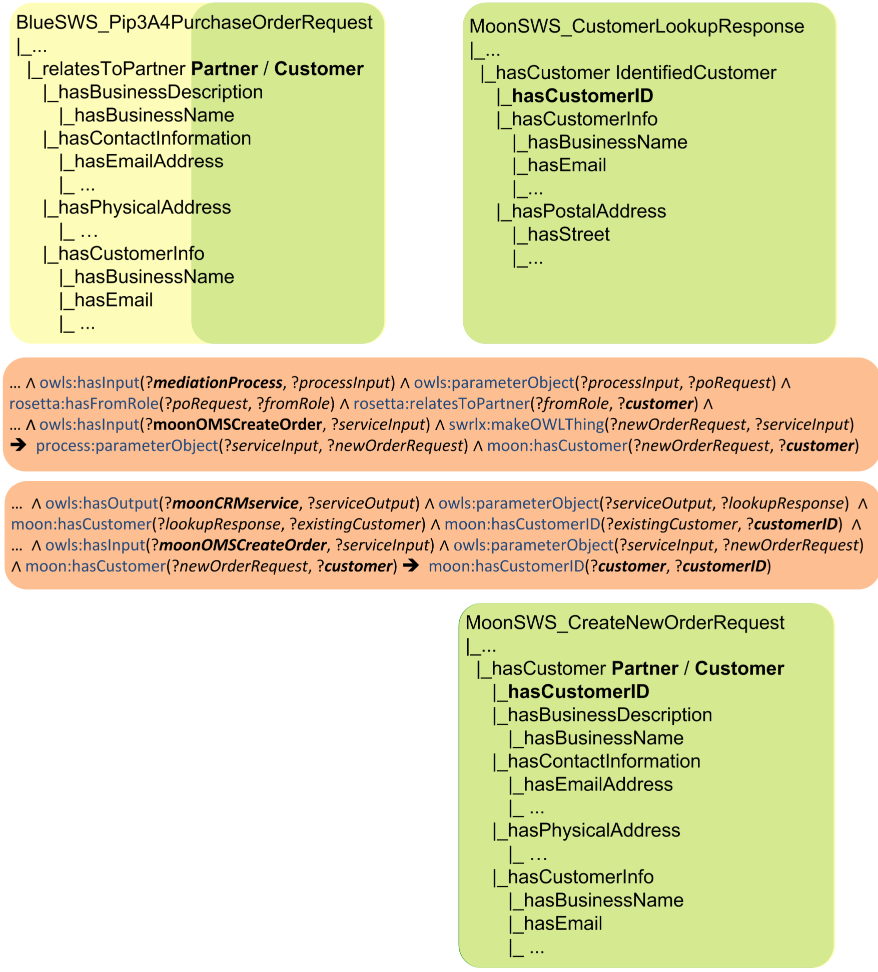


**Fig. 4.** Rule-based data flow example

### 2.5 BPEL-process for mediating interaction patterns

As already outlined above, the harmonization of the interaction patterns of both systems is achieved by means of a BPEL process definition and an corresponding BPEL engine [13].

The following example demonstrates how BPEL is combined with the rule-based data flow in order to mediate between different granularity levels in service calls. In the given scenario the *OrderItems* provided by the Blue system are aggregated in a single incoming call. However, the Moon system requires fine granular calls of the *AddLineItem* Semantic Web service, i.e. one service call for each single LineItem. Fig. 5 illustrates the corresponding BPEL-process part.

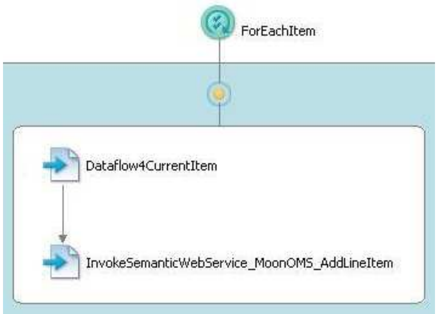


**Fig. 5.** BPEL-process part implementing the split of aggregated *OrderItem*s and the invocation of the *AddLineItem* Semantic Web service provided by the Moon Order Management System.

The *ForEachItem*-loop matches each single *OrderItem* and extracts its URI from the XML serialization defined according to the RosettaNet ontology. The following XPath expression is used for this purpose.

```
($processInputLifted//rdf:Description/rdf:type[@rdf:resource=
                        … MoonOntology.owl#OrderItem'])[$counter]/../@rdf:about
```

Subsequently, the URI is passed to the description logic based data flow component, which performs the actual data flow to the input variable of the Moon order management system, i.e. to the *AddLineItem* Semantic Web service. The URI of the *OrderItem* was chosen as the connecting conceptual element between the XSD-based data model of the BPEL process and the description logic based data model of OWL individuals and SWRL rules. Fig 6. illustrates the above described data flow rule.

… ∧ rosetta:hasProductLineItem(?***purchaseOrder***, ?***itemURI***) ∧
… ∧ owls:hasInput(?*moonOMSAddLineItemService*, ?*serviceInput*) ∧
owls:parameterObject(?*serviceInput*, ?*addLineItemRequest*)
→ moon:hasItem(?***addLineItemRequest***, ?***itemURI***)

**Fig. 6.** Rule-based data flow definition preparing the input of the Moon order management system, i.e. of the *AddLineItem* Semantic Web service. The variable *?itemURI* is replaced by the actual URI of the *OrderItem* extracted by the XPath expression at runtime.

The enhancements for applying the rule-based data flow and the Semantic Web service call have been implemented using the same standard BPEL extension mechanism as explained above.

## 3 Conclusion, Potential and Limitations

The approach presented in this paper aims at reducing the complexity of semantic mediation in order to facilitate system integration in a service-oriented architecture landscape. Semantic Web technologies have been used as an additional layer on top of existing WSDL-based Web services and XML Schema based message formats. The heterogeneous information models of the Blue system (RosettaNet) and the Moon system have been additionally expressed in terms of two autonomously developed domain ontologies. These domain ontologies have been combined with specific upper ontologies for Web services (OWL-S) in order to realize the Semantic Web services. Future versions of the presented implementation will be based on SAWSDL service descriptions. A corresponding Java API for the execution of SAWSDL Semantic Web services is currently under development by the authors of this paper.

The approach of multiple independent ontologies on the domain level takes into account the realistic point of view that in a system integration scenario the availability of a single global ontology covering the needs of all autonomous actors and organizations across various domains cannot be assumed. The well established paradigm of loose coupling is reflected on the semantic level by using independently evolving information models (ontologies) that are loosely coupled through semantic bridges.

While on the one hand semantic bridges target the semantic mediation challenge of heterogeneous data formats, the semantic mediation challenge of heterogeneous interaction patterns has been addressed by using a BPEL engine as the coordinating entity. The integration of Semantic Web services, semantic bridges and rule-based dataflow into BPEL processes thus provides the foundation for powerful tool support in semantic system integration scenarios. The description logic-based data model provided by ontologies in conjunction with semantic bridges allows for applying automatic semantic matching mechanisms based on polymorph representations of service parameters. In [1] this approach has been demonstrated in order to provide semantics-based tool support for Web service orchestration: Semantically matching service parameter parts are presented as assignment recommendations to the process expert, thus facilitating the data flow design. Based on the selection of the involved Web services and of the assignment recommendations an execution plan for the Semantic Web service composition can be constructed in a semi-automated manner. By applying the mechanisms described in this paper the generated process can be expressed in BPEL and executed by appropriate industrial mature workflow engines.

Thus, the presented approach relies strongly on the existing Web standards BPEL and OWL and thus raises the claim of being not only of theoretical but also of practical relevance. The main conceptual advantage of the presented approach is that semantic interoperability is addressed on the level of domain standards assumed as given in terms of ontologies. Thus, mediation between different representation formats in overlapping conceptualizations is only done once instead of performing it repeatedly on the application level during Web service composition. Consequently, the process expert can focus on process specific concerns and can leave the task of semantic mediation between heterogeneous information models to the domain experts. The task of semantic mediation between different interaction patterns is supported by providing semi-automatic tool support for data flow design. Thus, the complexity in semantic system integration can be reduced substantially.

## 4 Future Work

Future work will focus on exploiting the introduced semantic layer (Semantic Web services, semantic bridges, rule-based data flow) for further extension of tool support in semantic system integration scenarios in the context of service oriented architecture landscapes. The long-term vision on this regard is to reduce technical complexity for process experts in system integration and thus fulfill the conceptual promise of service oriented architectures namely to enable sound alignment of business with information technology.

**Acknowledgments.** This work is partially supported by the European Commission through the research project QualiPSo (Quality Platform for Open Source Software, contract IST-FP6-IP-034763). The paper was developed in the context of the "semantic interoperability" work package of the QualiPSo project.

## References

1. Nils Barnickel, Matthias Fluegge, Kay-Uwe Schmidt: Interoperability in eGovernment through Cross-Ontology Semantic Web Service Composition. Proceedings Workshop Semantic Web for eGovernment, 3rd European Semantic Web Conference 2006, Budva, Montenegro (2006)
2. Web Service Modeling Ontology (WSMO), Submission request to W3C, June 2005
3. Semantic Web Service Framework (SWSF), W3C Member Submission July 2005
4. Ressler et. al.: Application of Ontology Translation, Proceedings of the Sixth International Semantic Web Conference (ISWC2007), Pusan, Korea, 2007
5. Ehrig, M.; Staab, S.: QOM - quick ontology mapping. In Proc. 3rd ISWC, Hiroshima (JP), 2004
6. Armin Haller et .al.: Handling heterogeneity in RosettaNet messages, Semantic Web and Applications, SAC 2007, Seoul, Korea, 2007
7. W3C Web Ontology Language OWL, http://www.w3.org/Submission/2004/07/
8. Evren Sirin, OWL-S API, Maryland Information and Network Dynamics Lab Semantic Web Agents Project
9. W3C Semantic Annotations for WSDL Working Group, http://www.w3.org/2002/ws/sawsdl/
10. W3C Semantic Web Rule Language SWRL, http://www.w3.org/Submission/SWRL/
11. Klein M. et. al.: The Relation between Ontologies and Schema-languages: Translating OIL-specifications in XML-Schema. In Proceedings of the 14th European Conference on Artificial Intelligence ECAI'00, 2000
12. Protégé-OWL, http://protege.stanford.edu/overview/protege-owl.html
13. OASIS Web Services Business Process Execution Language Version 2.0, http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf