Dynamic Data Mediation in Enterprise Application Integration

Thanassis Bouras\(^1\), Panagiotis Gouvas\(^1\), Gregoris Mentzas\(^1\)

\(^1\) Institute of Communication and Computer Systems, National Technical University of Athens, 9, Iroon Polytechniou str., Zografou Campus, Zografou, Athens 15780, Greece.

Tel: +30 210 7723895; Fax: +30 210 7723550;
Email: \{bouras, pgouvas, gmentzas\}@mail.ntua.gr

Abstract: If we try to increase the level of automation in Business-to-Business Enterprise Application Integration (EAI) scenarios, we confront challenges related to the resolution of data and message heterogeneities – that traditional, syntactic EAI technologies are weak to solve, as they miss the documentation of the semantics related to the interfaces and the data structures of the participating services. In this paper, we propose a semantically-enriched approach for dynamic data mediation in EAI scenarios, based on Ontologies, Semantic Web and Semantic Web Services Technologies. The proposed approach focuses on the resolution of message level heterogeneities between collaborative enterprise services exposed from the participating business systems, facilitating automatic, dynamic data mediation during execution time by providing formal transformations of the input and output messages (of the participating Web Services) to a common reference model, i.e. the Enterprise Interoperability Ontology. Moreover, we present a tool that has been designed and developed to support the user to provide business data-related semantic annotations and XSLT transformations if the input and output message parts of given Web Services exposed from business applications, realizing parts of their functionality. Finally, we demonstrate the utilization of the proposed approach and toll in a real-world EAI scenario, i.e. the Stock Replenishment process, across a franchisor-franchisees collaborative value network.

Keywords: Semantic Web, Ontologies, OWL, Data Semantics, Data Mediation, XSLT Transformations, Semantic Web Services, SAWSDL, Enterprise Application Integration.

1. Introduction

In the mid-1990s, a new term called enterprise application integration (EAI) was established, which introduced several methods and software components for efficiently integrating software in an enterprise. Since then available enterprise application integration solutions address integration problems in the (indicative) following ways [1]: by graphically supporting the mapping of systems’ interfaces to each other (e.g. SAP NetWeaver Exchange Infrastructure); by reducing complexity using intermediate data-exchange languages (e.g. Extensible Markup Language - XML) or by reducing the number of connection adapters needed through the introduction of hubs (e.g. Enterprise Service Bus). These efforts entail significant costs and typically due to the “lack of automated support in defining integration, it takes a long time for a human engineer to define semantically correct integration” [2].

The problem that still exists, which the traditional, syntactic EAI technologies are weak to solve, refers to the formalization and the documentation of the semantics related to the interfaces and the data structures of the deployed Web Services. This lack of formal semantics of applications and services to be integrated makes it difficult for software
engineers and developers to interconnect heterogeneous applications and thus creates obstacles in the automating EAI activities [3].

There is no doubt that these needs impose the use and interpretation of semantics in EAI and that semantically enriched approaches will hopefully mitigate these problems.

In this paper, we present a semantically-enriched approach for dynamic data mediation in Enterprise Application Integration scenarios, based on Ontologies, Semantic Web and Semantic Web Services Technologies. Our approach, which is presented in the next section, focuses on the resolution of message level heterogeneities between collaborative enterprise services exposed from the participating business systems, facilitating automatic, dynamic data mediation during execution time by providing formal transformations of the input and output messages (of the participating Web Services) to a common reference model, i.e. an enterprise data ontology. In addition, the next section provides an overview of the enterprise data ontology that we have developed and utilized as part of a multi-layered and –faceted interoperability ontology, called Enterprise Interoperability Ontology, which provides a shared, common understanding of data, services and processes within enterprise application integration scenarios.

In Section 4 we present SEAP, the Semantic Annotation and Profiling tool that we designed and developed in order to support the user to provide business data related semantic annotations to specific web services exposed from enterprise applications. The SEAP tool enables the user to graphically define the required transformations of the output and input messages between web services with regard to the respective data entities (used for the annotation of these message parts) of a common ontological model. These transformations are further utilized to enable dynamic data mediation among several interconnected enterprise services, during the execution of a business process which contains these services.

Finally, we provide (in Section 5) an indicative business scenario demonstrating how our proposed approach and tool contributes to the resolution of data heterogeneities among different business, while, in Section 6, we summarize the conclusions and the future work of the research efforts presented.

2. Semantically-enriched Data Mediation

As already stated the proposed data integration approach facilitates automatic, dynamic data mediation during execution time by providing formal transformations of the input and output messages (of the participating Enterprise Services) to a common reference model (an enterprise interoperability ontology that we developed in OWL, the Web Ontology Language).

More specifically, automatic, dynamic data mediation is enabled by providing a priori mappings and transformations for all enterprise services inputs and outputs message parts (of the services’ native Web Services) to a common-reference conceptual, ontological model, i.e. the data-intensive enterprise interoperability ontology.

Mappings are created between the enterprise services message elements and ontology concepts, utilizing the schemaMapping attribute to semantically annotate and associate the input and output message elements of the involved enterprise services, towards the creation of the so-called Semantic Profiles of these services, respecting the SAWSDL specification for the deployment of Semantic Web Services [4].

In the literature [5], two types of mappings between enterprise services message elements and semantics have been identified: a) mappings from the Web Service message element to the ontology concept, also called the “up-cast” and/or “up-level”, and b) transformations from the ontology concept to the message element, called the “down-cast” and/or “down-level”.
Once these transformations are defined, two enterprise services can interoperate by reusing these mappings. Both the mappings and the message transformation occur at the instance level between the WSDL (XML) and the OWL individual.

2.1 Extending Native Web Services Interfaces with Data Semantics

As already mentioned above, the main idea behind the proposed dynamic data mediation approach is that the native descriptions (i.e. WSDL interfaces) of the standard Web Services of the involved service-oriented business applications is extended with data/information semantics. This extension process is called semantic annotation/profiling and the resulting description is the respective SAWSDL-compatible Semantic Profile.

It is obvious that, in order to solve real integration use-cases, the composed processes must also be able to execute/run. Consequently a special start event must trigger the runtime engine to create a process instance. The process instance works with real data which means that it communicates with real Web Services of SOA-enabled business applications. As a result, the process instance includes an invocation of one Web Service with given input data, stores the reply of this Web Service, invokes another Web Service with the stored data, and evaluates logical expressions to decide which execution branch to follow according to stored data, etc.

Because the process instances communicate to the outside world by using the semantic concepts of the common ontological reference model, the standard Web Services of the service-oriented applications can’t understand them directly. To solve this, the standard Web Services are encapsulated (through the semantic annotation/profiling process) into semantically-annotated Web Services, which are directly invokable by a given process instance. These services will be referred as “mediated services”.

In the next section we introduce a typical semantic annotation/profiling scenario.

2.2 Walkthrough of Enterprise Services Profiling

We assume that Web Services of the service oriented business applications – called native Web services – are described by a WSDL file. This WSDL description contains the data structure (or a reference to it) of the requested input and the provided output data of operations the service provides. This structure is in the format of an XML schema definition (XSD) and is called native data structure. During the annotation process, the standard WSDL description will be extended with semantics as described in the following paragraphs.

The exact connection between the native data structure and the used ontological concepts has to be defined first. The concepts of the data-intensive interoperability ontology (i.e. the common reference model), which are the closest to native data, are identified and referenced from the extended description of the native Web Service. This results in a semantic description which is constructed by the guidelines of the SAWSDL recommendation, i.e. creation and utilization of a “modelReference” annotation mechanism pointing to a data ontological concept, for every wsdl:part of the involved Web Services.

Although one can find concepts which are very close to the native data in meaning, the syntactical differences between the concepts have to be bridged by creating and applying XSLT transformations.

The XSLT transformations are also used to handle data mismatch problems. Data mismatch problems may be caused from differences between native data structures and the selected concepts of the enterprise data ontology. This can be due to the usage of different units (e.g. when a native service uses US units whereas the Ontology concepts use metric units), different currencies, different format for the same data (for example the address, or
the date is formatted in many different ways in CRP and ERP systems due to the differences between national traditions).

The created XSLT transformation code is stored at a common repository. After the transformations are present, the reference to them is added to the semantic description, i.e. the Semantic Profile, of the service.

3. Overview of the Enterprise Interoperability Ontology

Our approach for dynamic data mediation is based on the definition of formal transformations of the input and output messages of the involved Web Services to a common reference model. For this purpose, we are utilizing the Data Facet of the Enterprise Interoperability Ontology (ENIO) that we have designed and developed [6].

The Enterprise Interoperability Ontology represents an explicit specification of the conceptualization of the EAI domain, and structures and formalizes the procedural and operative knowledge needed to describe and resolve the given EAI problem, providing a formal and explicit definition of the data, services and processes that exist within an application integration problem.

Regarding the design principles for constructing ENIO, we have chosen to introduce the model of an upper ontology, which covers generic and domain-independent concepts, with several, domain-related extensions that we call facets. We have developed a three-faceted structure for ENIO: data facet; functional facet and process facet. As the Data Facet of ENIO aims to formally capture the semantics of messages exchanged among collaborative enterprise applications that expose their functionality as web services, it fits perfectly within the scope of the proposed semantically-enriched data mediation approach.

As we do not intend to re-invent the wheel, we based the ENIO Data facet on the Core Components Technical Specification (CCTS). CCTS is currently the ISO 15000-5 Technical Specification and is supported and used by more than 50 projects and initiatives (including UBL, RosettaNet, CIDX, SWIFT, OAG, etc). The meta-model of the ENIO Data Facet ontologizes the meta-modelling elements of CCTS, i.e. Core Components (CC), Data Types (DT), Aggregated CC (ACC), Basic CC (BCC) and Association CC (ASCC).

For the population of the Data Facet, we have utilized as knowledge sources the following standards and vocabularies: the OASIS ebXML Core Components Dictionary, the RosettaNet Business Dictionary, the OAGIS specification, and the OASIS Universal Business Language.

Finally, as our goal is to provide a generalized reference ontology for a semantically-enriched data mediation within EAI scenarios, a fine-grained axiomatisation is not needed. A semi-formal ontology providing a common vocabulary with a formal taxonomy, but without detailed logical axioms is enough for our purposes. We therefore chose a common denominator of ontology features which are present in all current ontological formalisms, including but not limited to RDFS, OWL, and WSML.

The features we use are the following: concepts with formal sub-concept relation, instances with formal instantiates relation, and binary properties with single concept domain and range constraints. We have chosen OWL-DL as our implementation language because it is an already available W3C recommendation and has good tool support.

Moreover, other ontology formalisms (including WSML) provide conversion utilities from and to OWL-DL. We therefore expect that ENIO can be (semi-) automatically translated into various other formats in the future, if it is needed by the target application domain.
4. Semantic Annotation and Profiling of Enterprise Services

We developed SEAP (a Semantic Annotation and Profiling tool) which allows the selection and visualization of the native Web Services interfaces and the data-intensive enterprise interoperability ontology and facilitates the user to annotate (through a drag and drop utility) the input and output message parts of the selected Web Service interface with business data entities of the already mentioned enterprise interoperability ontology.

As shown in Figure 1, the graphical user interface of SEAP comprises of five main areas. In the left side, we have the Native Services Area that is responsible for the visualization of the selected, native Web Services that are exposed from the existing business system of an enterprise. In the middle, we have the Ontology Browser area that is responsible for visualizing all the tree facets of ENIO, i.e. the Functional, the Data and the Behavioral one, as well as, the Ontology filtering area, which sorts out the visualized ontological concepts, based on given keywords. Finally, in the right-hand side, the Adaptation Layer Services area visualizes the mediated services that comprise the respective collaborative business process, and the Adaptation Layer Concepts area, which contains the ontological concepts that annotate inputs and outputs of the services involved in the respective collaborative business process.

The user is allowed to drag and drop ontological concepts of the ENIO Data Facet visualized in the Ontology Browser Area, annotating the input and output message elements of the native WSDL interface visualized in the Native Services Area. In parallel, the adaptation layer services and concepts are automatically created. Once the annotation procedure is completed, the user is guided to provide and graphically define the respective XSLT transformations from (for input messages) and to (for output messages) the Enterprise Interoperability Ontology. The up- and down-casting transformations are stored along with the data semantics annotations to the SAWSDL-compatible semantic profile of the selected service. Thus, these semantic extensions (i.e. data annotations and XSLT transformations) are further utilized during the execution of generated collaborative business process models that interconnect and orchestrate already annotated enterprise services.
5. Demonstration with a Business Case

Our demonstrator is based on a typical franchisor-franchisees value-added network which includes a complex IT infrastructure in the franchisor headquarters comprising of several centralized, corporate systems, i.e. ERP, CRM and WMS, which are required for the coordination of the retail activity, reimbursements, logistics and the pricing policy of the chain of retail stores. On the other hand, the Point-of-Sales (PoS) retail stores are equipped with an ERP-like Retail System that allows the collaboration with the franchisor and facilitates the business activities of the value network.

In this enterprise context, we have identified several Collaborative Business Processes (CBPs) that compose and invoke (complex) services exposed from heterogeneous business systems. We have selected the “Stock Replenishment” CBP, in order to demonstrate the usage and applicability of the proposed approach in facilitating dynamic data mediation in complex enterprise application integration. The daily store replenishment cycle constitutes a typical case of process integration scenarios. This replenishment procedure imposes that every day the franchisees have to replenish their stocks by ordering new items from the franchisor that has to invoice and deliver the items requested.

The replenishment procedure involves and triggers, in all way, processes of most of the systems comprising the franchising network infrastructure. We are going to demonstrate the
proposed approach and tool through the exchange of the “Address” data object (Figure 2) among the participating services of the involved business systems.

Utilizing the Semantic Annotation and Profiling (SEAP) tool, the user (after having selected and visualized both the native WSDL of the participating Web Service, containing the “Address” object, and the data facet of the Enterprise Interoperability Ontology) annotates the respective message part with the “DataFacetsV3:Address” term existing in the Enterprise Interoperability Ontology (Figure 3).

Once, the data-related semantic annotation of the “Address” object message part is completed, the user interacts with the “XSLT Transformation Editor” of the “Enterprise Services Semantic Profiling” tool so as to graphically define all the required mappings and transformations from the structure of the native “Address” data object to the respective ontological term (Figure 4).

The developed XSLT transformations are stored along with the data semantics annotations to the SAWSDL-compatible semantic profile of the selected service (Figure 5). These XSLT transformations are further utilized during the execution of generated
collaborative business process models that interconnect and orchestrate already annotated enterprise services.

6. Conclusions

In this paper we proposed a semantically-enriched approach for dynamic data mediation among interconnected Web Services in the framework of Collaborative Business Processes. Moreover, we presented a Semantic Annotation and Profiling (SEAP) tool developed to support the end-user to semantically annotate (with data semantics) the Web Services interfaces of the exposed enterprise services and to graphically define the required up- and down-casting XSLT transformations.

In addition, we presented the applicability of the proposed approach in a realistic B2B integration scenario within a franchisor-franchisees collaborative value network, comprising complex, heterogeneous systemic infrastructure. In the frame of this scenario, we have demonstrated the dynamic resolution of data heterogeneities at execution time.

In the future, we intend to generate and add a transformation repository, containing all the supported transformation types, which can be selected and reused during a service’s semantic annotation and profiling process.

Acknowledgement. Work presented in this paper has been partially funded by the European Commission in the project FUSION (Business Process Fusion based on Semantically-enabled Service-Oriented Business Applications) IST 027385.

References