ONAR: AN ONTOLOGIES-BASED SERVICE ORIENTED APPLICATION INTEGRATION FRAMEWORK

Dimitrios Tektonidis[1], Albert Bokma[2], Giles Oatley[2], Michael Salampasis[3]

[1] ALTEC S.A., Research Programmes Division, M.Kalou 6, GR – 54629, Thessaloniki, Greece
dte@altec.gr

[2] Centre for Electronic Commerce, University of Sunderland, Sunderland, SR6 0DD, United Kingdom
Albert.Bokma@sunderland.ac.uk;Giles.Oatley@sunderland.ac.uk

[3]Department of Informatics, Technological Educational Institute of Thessaloniki, Thessaloniki, Greece,
cs1msa@it.teithe.gr

Abstract. The evolving technologies of Semantic Web and Web services are providing new means for application integration frameworks. The need for semantically enriched information exchange over the flexible environment of the internet provides a valuable enhancement to traditional methods and technologies for Enterprise Application Integration. However the utilization of the Semantic Web and Service Oriented Architecture (SOA) is not as straightforward as it appears and has specific limitations and inefficiencies due to the fact that is was not originally designed for that purpose. This paper presents a methodology that aims at the exploitation of these two technologies and the definition of an ontologies based enterprise application integration framework (ONAR).

1 Introduction

The application integration between heterogeneous applications such as ERP systems between companies as well as B2B environments in general is a challenge that has been confronted with a plethora of technologies and methodologies. In our research [1] we have seen that the key aspect in from simple integration to more complex cases [2] is the semantics. When we refer to the semantics we signify the meta-information that describes the exchanged information.

Service oriented application integration (SOAI) as it is presented in [3] exploits the capabilities for the functional description of web services that are used for the actual integration. This paper presents our work on the creation of an integration framework based on SOAI that utilizes Semantic Web Technologies [4] in order to enrich the semantics of the exchanged information.

The Ontologies Based Enterprise Application Integration (ONAR) Framework [5] utilizes web ontologies to create semantic conceptualizations of the business concepts that exist inside an application. This conceptualization is used for the creation and the registration of the Web services in a UDDI based registry.

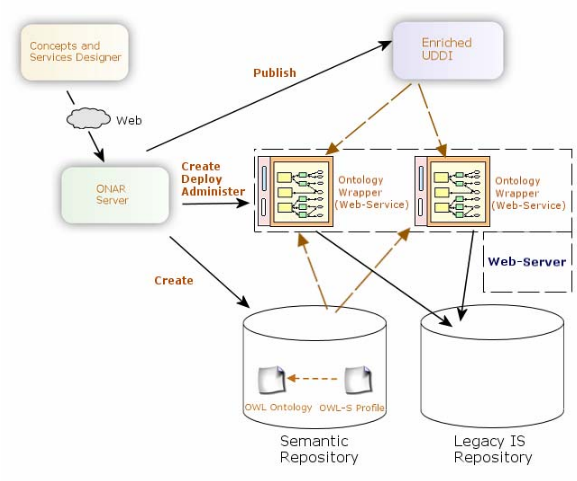

Fig. 1. Ontologies Based Enterprise Application Integration (ONAR) Framework.

2 Conceptualizing Information System using OWL Ontologies

Ontologies are used for the formal definition of concepts in a given domain. However the utilization of ontologies for software component definition is not as simple as the definition of concepts alone. Research on this domain as presented in [6] and [7] shows that the utilization of the ontologies requires some modification in the principals of the frame-based ontologies languages. The state of the art ontologies languages are presented evaluated in [8] and [9], but even a purely frame-based logic is still insufficient as there are important differences principally with respect to object-oriented modeling. Of course ontologies were not designed for software engineering, and thus their utilization for this purpose requires creating a semantic layer that will describe and define the technical infrastructure of an application.

Our work has focused on creating associations between the conceptional models and the application logic. To facilitate this we have developed a graphical tool (fig. 1) that enables the creation of the conceptual models using ontologies and associating these models to the actual system resources.

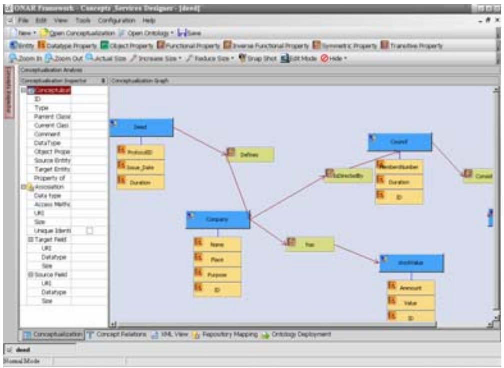

Fig. 2. Ontologies based Information System conceptualization using ONAR Concepts and Services Designer

2.1 Definition of Logical Entities

Following the OWL [10] frame based logic, the concepts that represents a logical entity inside the application are defined by an OWL class. The entities inside the ONAR conceptualization methodology inherit all the features of the OWL classes. However their semantics changes in order to be adapted to the semantic description of a system resource.

Inheritance is permitted allowing some class features to be inherited from one class to another. The relation between the parent class and child classes follows the same-rules that both object oriented engineering and frame-based logic supports, however multiple inheritance is not permitted. This constraint is due to the fact that in complex information system polymorphism of some concepts will unduly increase the complexity of conceptualization.

In our approach, the conceptualization of the system is not application centric but integration centric. This means that the creation of a conceptualization is developed and the need of the integration and not regarding the structure of the system. This increases the reusability of the concepts definition that can be reused in different cases.

Entities are allowed to have properties that correspond to OWL datatype properties. A property can belong to more than one entities and is contrary to the usual principles of object oriented modeling. This principle derives from the frame based logic of OWL ontologies and extends the object oriented engineering where a property belongs to only one class.

Finally both entities and properties are enriched with notional properties like lexical descriptions, the ability to define the inverse of a concept and the ability that a property can potentially be derived from another property. These enrichments are used to increase the inference for the semantics of the concepts and their properties inside a model following the OWL principles.

2.2 Defining Relations between Entities

Apart from the definition of entities and their properties, our work has focused on creating semantics in the relations between concepts. The existing semantics of relations that are expressed in object oriented class diagrams in UML or Entities Relations (E-R) diagrams pose limits to the semantic two concepts may have at the logical level. Therefore using the four types of relations (object relations) that are defined in OWL we have adjusted these relations types in order to increase the inference capabilities of the conceptualizations. We have followed the OWL logic and syntax having the two (or more) entities that participate in a relation divided into:

a) **Domains** of the relation: that is consisted of the concepts that consist of the area of definition of the relation.
b) **Ranges** of the relation: that the domains concepts of the relation can receive values from.

More precisely we have used and adjusted the following relations between entities:

- **Functional** relation: where the domain concepts instance can be related to more than one instance of the range concepts.
- **Inverse Functional** relation: where the range concepts instances can be related to more than one domain concepts instances.
- **Symmetric** relation: 1 to 1 relation between the instance of the domain and the range concepts.
- **Transitive** relation: we have defined transitive property as an inferential symmetric property to facilitate the logical allocations of concepts.

However we have to limit the syntax of OWL which permits the existence in a relation of more than one concepts as ranges and domains. In our approach (in a relation) if the domains contain more than one concept then the range should be defined from only one concept and vice versa. This ensures that in the association of the logical model to the system resources will not create indeterminism.

3 Associating logical entities to information system repository

One of the most important results of our work is the association between the logical model and the repository of the system. In our early attempts we have tried to extend OWL in order to include mapping mechanisms. However we have discovered that the new “extended OWL” syntax will be too complicated and it will mix up semantic with syntactic issues. This of course would decrease the reusability of the conceptualizations.

Therefore we distinguish the conceptualization into two ontologies that are:

a) The **conceptualization** ontologies based on OWL that define semantics aspect of the conceptualizations.

b) The **association** ontologies that association the conceptualization ontology to the repository of the application.

The disassociation between the semantic and the syntactic description has as the following results:

a) To increase the **reusability** of the conceptualizations ontologies: The same conceptualization ontologies can be used to different information system.
b) To reduce cost of **maintainability** when changes happened in to the application repository.
c) To enable the creation of **common vocabularies** that can be shared between heterogeneous applications.

In our work so far we have achieved the application of the ONAR framework to relational databases but where the goal is to support any type of repository including application servers.

4 SOAI using semantic conceptualizations

The creation of a Service Oriented Architecture that will implement the integration between the applications is also based on the semantic conceptualizations. In fact the ONAR framework software components (Web services) are based on the conceptualization schema (the set of the conceptualizations) of the application. This strong relation between the conceptualizations and the web services server is the actual innovation compared with other existing approaches [11][12].

The ONAR integration process is the methodology that should be followed in order an integration case to be defined and implemented using the ONAR framework. The process consists of the following phases:

a) **Conceptualization phase** where the entities that need to participate in the integration case are conceptualized to shared conceptualizations (OWL ontologies). The set of the conceptualizations create the conceptual schema of the integration.
b) **Association phase**. The conceptual schema is associated to the information systems repositories that participate in the integration.
c) **Design phase**. In this phase using the conceptual schema the definition of the software instances takes place.
d) **Deployment phase** where the software instances are implemented as software components (Web services) and deployed.
e) **Publication phase**. The framework uses a UDDI registry enriched with OWL-S profile features (from semantic invocation) to publish its software instances. Using the semantic meta-information that the conceptual schema provides the framework publish information regarding the utilization of the web services.

4.1 Designing web services based on the conceptual schema

In our work, the whole web services design creation and their registration to the UDDI registry is related to the conceptual schema. The design of the software instance is the process where elements of a conceptualization are used to define the input and the output of the Web service. The design of a web service consists (figure 3) of its syntactic definition and its semantic description.

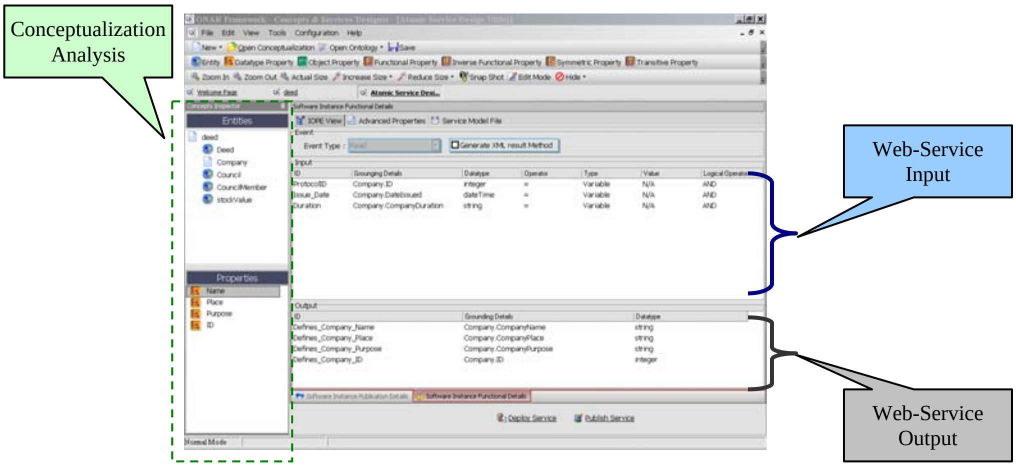

Fig. 3. Designing Web services using a conceptualization Ontology

The syntactic definition defines the input and the output as well as the behavior of the Web service. This definition is based one conceptualization of the conceptual schema. Starting from one basic entity the Web service designer can use all the concepts that are necessary for his definition. The syntactic definition (ONAR Service Model) of the service contains all the semantic and syntactic relations between the basic entity (primary concept) and the secondary concepts. The **ONAR Service Model** [5] is an XML document that contains a conceptualization ontology and an association ontology of the functionality of a Web service.
The reason of having a new definition document for the syntactic definition of the service, apart from maintenance purposes, is to increase the portability of the web services. We have implement the necessary functionality that enable us to automatically generate the source code (C#) of a Web service based only to the conceptualization ontology the association ontology and the ONAR Service Model. Therefore if two applications share the same conceptualization ontologies but have different structures (different association ontologies) they can exchange Web services.

4.2 Creating semantic description for supporting SOAI

The Web Service Definition Language (WSDL) is used to describe web services at the technical level. However the use (or consumption) of a web service requires first

the technical implementation and an understanding of its functionality. The capabilities of the WSDL at this level are limited as it was not designed to serve this purpose. Instead other protocols have been produced in order to support Web services discovery and description.
Our work has been based on M.Paolucci attempts [13][14] to integrate a powerful UDDI discovery and categorization mechanisms with the semantics of OWL-S [15]. We have extended UDDI using OWL-S elements in order to include semantic description of the web services. Apart from the ONAR service model that a service designer defines with the definition of the behavior of the web service (Inputs, Outputs and effects), the designer should describe the web service behavior at the semantic level.

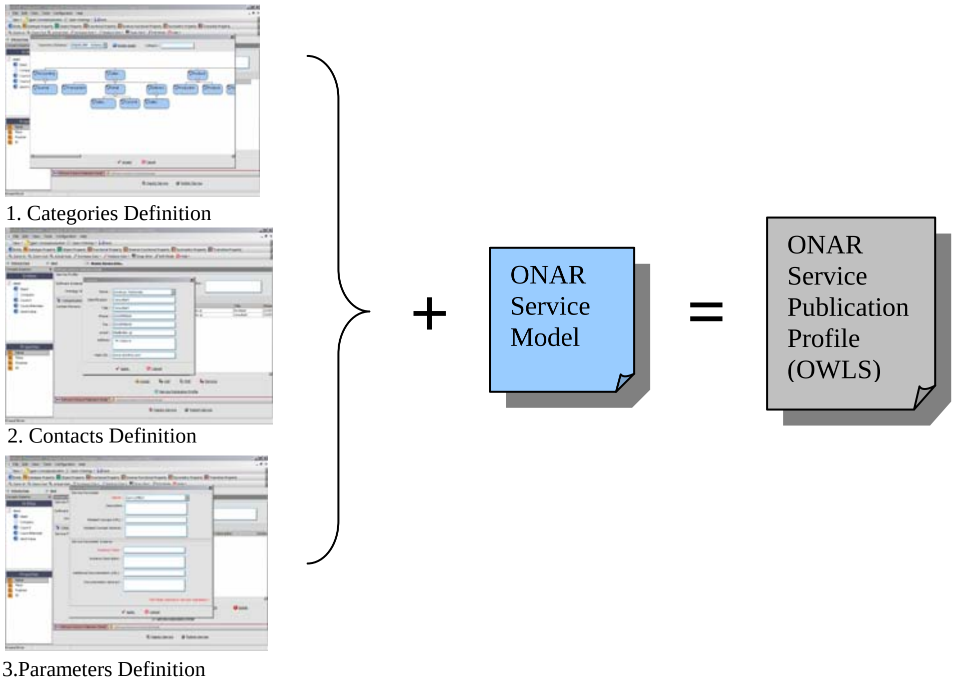

Fig. 4. ONAR Service Publication Profile

This Service Publication Profile defines the semantic of the web services. The Web service designer defines:

1. The **categories** of the service: Using the UDDI categorization schemas the web service designer defines the categories that apply to the web service.
2. The **contact list** of the service: A list of persons that are associated with the service and their type and responsibilities

3. The **parameters** of the service: Special features of the services that do not belong to a particular categorization schema.

Inside the Service Publication Profile we have also included the elements of the Service Model (Inputs, Outputs and Effects). These elements that are parts of certain conceptualizations carry also semantic descriptions that are also registered to the UDDI registry.

These semantic descriptions of the web service leverage the semantic discovery of the web services reducing the errors of incorrect usage of the web services. This is considered essential for effective SOAI solutions regarding the number of the web services required to support complex integrations and the complexity of each Web service.

5. Extending UDDI registries to support semantics

In order to include semantic meta-information to UDDI registries we have extended both the UDDI and OWL-S profiles. Our work was based on M.Paolucci work in this domain and also some recommendations from [16]. The ONAR Service Publication Profile is based on OWLS profile syntax with some additions that can be summarized (further information can be found on [5]) as follows:

a) Enhancement of the OWL-S input and OWL-S output class in order to define the related classes and relations.
b) Enhancement of the OWL-S Service Category and Service Parameter is order to exploit all the UDDI description capabilities
c) Additional Models to support OWL-S concepts like Input, Output, Effect, Actor and Service Parameter

This integration between the UDDI and OWL profile leverage the search capabilities that can be based on concepts of conceptualizations. We have developed a search facility (figure 5) that enables the user to choose semantic filters in the level of class properties or relation between classes apart from the typical UDDI inquiry. The search can use both of the techniques lead to more accurate results.

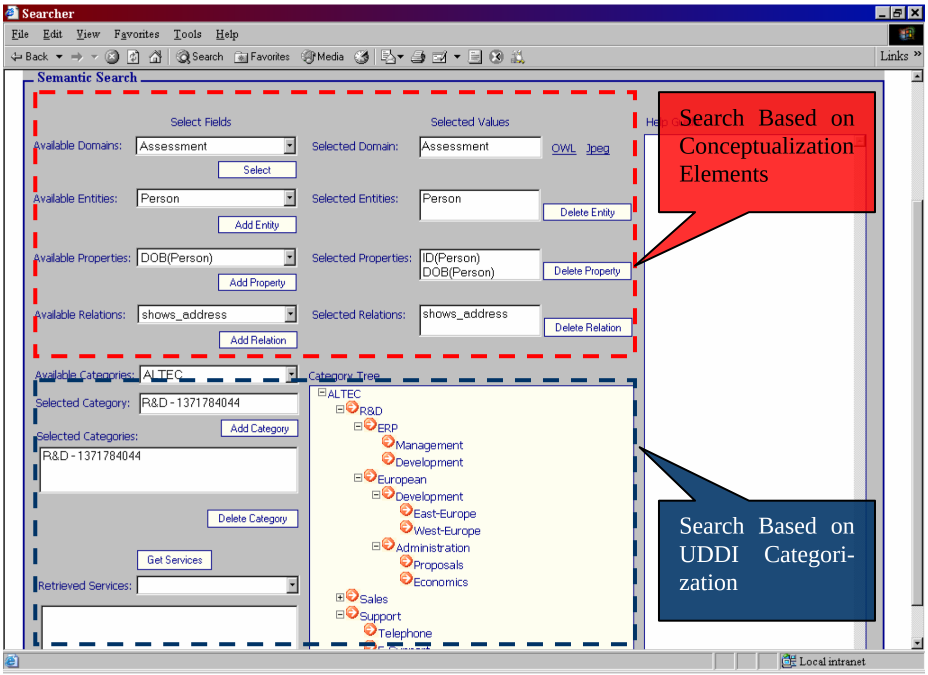

Fig. 5. ONAR Service Publication Profile

6. Conclusions and future work

The efficient handling of a large number of web services and the expressiveness in the description mechanisms that define the functionality of the web services are considered the major problems in SOAI. Our approach based on a common vocabulary that consists of web ontologies presents a solution that has been applied in a series of projects[5] and improves the efficiency than ad hoc SOAI.

Our framework provides a set of administration mechanisms [5] that facilitate the manipulation of the web services. These software components provide overall monitoring of the web service in order to reduce the cost of maintenance of the integration.

Based on the evaluation results of implemented test cases, we intend to enrich the expressiveness of the description mechanisms of the web service. Our experimentations focus on the improvement of the inference capabilities of the search mechanisms. In order to achieve this we are working on the integration between OWL-S profile and UDDI. The publication of the web services in our framework that is based on the conceptual schema should automatically publish the semantic meta-information that carries from their conceptualization ontologies.

Finally we are currently working on amending our web services search facility in order to exploit the semantic meta-information that has been published to the public registries.

References

1. D. Tektonidis, A. Vontas, U. Hess, J. Meschonat "Handling the Shop-Floor of an Industry through ERP systems – A functional Integration Model", International Conference on E-business & E-Work 2002, Prague 2002
2. M. Themistocleous, Z. Irani, R. O'Keefe & R. Paul "ERP Problems and Application Integration Issues: An Empirical Survey" Proc. of the 34th Hawaiian International Conference on Systems Science, Hawaii Big Island 7-10 January 2001, USA, IEEE 0-7695-0981-9/01
3. David Linthicum "Next Generation Application Integration: From Simple Information to Web Services", Addison-Wesley 2003, ISBN: 0201844567, USA
4. W3C, Semantic Web Description, 2001, available at http://www.w3.org/2001/sw/ , as accessed in 10 Nov. 2002
5. ONAR, Ontologies based Enterprise Application Integration Framework, 2004, available at http://research.altec.gr/ONAR , as accessed in 10 Oct. 2004
6. Li min Liu & M. Halper - "Incorporating semantic relationships into an object oriented database system"Proc. of the 32th Hawaiian International Conference on Systems Science, Hawaii Big Island 5-8 January 1999, USA, IEEE 0-7115-1405-9/99Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. 1 (1997) 108–121
7. Chia-Chu Chiang - "Wrapping legacy systems for use in heterogeneous computing environments", Information and Software Technology, Volume 43, Issue 8, July 2001, Pages 497-507
8. O. Corcho, M. Fernández-López and A. Gómez-Pérez "Methodologies, tools and languages for building ontologies. Where is their meeting point?" , Data & Knowledge Engineering, Volume 46, Issue 1, July 2003, pages 41-64
9. V. Ratnakar, and Y. Gil, "A Comparison of (Semantic) Mark-up Languages", Proceedings of the 15th International FLAIRS Conference, Special Track on Semantic Web, Pensacola, Finland , May 2002.
10. OWL 2003, W3C (2001b) OWL 1.0 Syntax Specification, available at http://www.w3.org/TR/owl-ref/ , as accessed in 15 May, 2003
11. M. Mecella & C.Batini "Cooperation of Heterogeneous Legacy In-formation Systems: a Methodological Framework" Proceedings of the 4th International Enterprise Distributed Object Com-puting Conference, Makuhari, Japan, 2000
12. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., & Hübner, S. "Ontology-based Integration of In-formation - A Survey of Existing Approaches", In: Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, USA, pages 108-117.
13. M. Paolucci, T. Kawamura, T. R. Payne, K. Sycara. "Semantic Matching of Web Services Capabilities", International Semantic Web Conference (ISWC), 2002, pp 333-347
14. M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara. Importing the Semantic Web in UDDI. In Proceedings of E-Services and the Semantic Web Workshop, 2002
15. DARPA, DARPA Agent Markup Language Web Services (DAML-S) Version 0.7 Specification, 2002, available at http://www.daml.org/services/daml-s/0.7/ , as accessed in 10 Nov. 2002
16. Dogac, A., Kabak, Y., Laleci, G., "Enriching ebXML Registries with OWL Ontologies for Efficient Service Discovery", 14th International Workshop on Research Issues on Data Engineering, Boston, USA , March 28-29, 2004.