An ARIS-based Transformation Approach to Semantic Web Service Development

Cheng-Leong Ang[φ], Yuan Gu, Olga Sourina, and Robert Kheng Leng Gay
Nanyang Technological University, Singapore
eclang@ntu.edu.sg[φ]

Abstract

This paper explores the use of ARIS for Semantic Web Service development. It proposes an ARIS-based transformation methodology for the automatic specification and development of Semantic Web Services. It also identifies research issues which have yet to be resolved for the development of a supporting software tool for the methodology.

1. Introduction

Web Services are emerging as a promising technology for inter-organisation integration. Recently, Semantic Web technology has also been introduced to make Web Services semantically interoperable, allowing them to be more dynamic through automatic service discovery, invocation, and composition. Semantically-rich languages such as OWL-S [12] have been created to provide a mechanism for describing the semantics of Web services. Unfortunately, for the common developers and users, the learning curve for such languages can be steep, creating a barrier for adoption and widespread use. To overcome the problem, we propose a transformation methodology for the automatic specification and development of Semantic Web Services (SWSs) based on ARIS [14]. The methodology centres on the creation of ARIS models and their automatic transformation to OWL-S specifications. The rationale is to combine the two to provide the best of both worlds, with ARIS meant for human users and OWL-S for computers. The methodology will enable automatic deployment on the Semantic Web of workflows that are described in ARIS. We target OWL-S here because it is likely to grow into the de facto standard for SWSs.

In this paper, we explain the methodology from the perspectives of a service requester, a service provider and a system administrator. We also identify research issues that have yet to be resolved for the development of a supporting software tool.

2. Motivation

SWS development using a transformation approach is nothing new. Narayanan and McIlraith [11] have used Petri nets to compose Web services. Bose et al [1] have suggested using UML for modelling OWL-S ontologies, while Dori et al [4] have proposed an approach based on wrapping OWL-S with OPM/S, a higher-level modelling language based on Object-Process Methodology (OPM) [3]. However, none of the existing transformation approaches is based on a suitable business process modeling language. Since Web services are basically business processes created on the Web, then a suitable business process modelling language should rightly be used for their descriptions. Neither Petri Net, UML nor OPM/S is meant for that; UML is actually meant for software engineering, OPM/S for general system engineering, and Petri Net for analysing the dynamics and control flow aspects of systems.

ARIS has been created specifically for business process modelling. It is a suitable business process modelling language because it provides a very comprehensive description of business processes (see section 3). Besides, ARIS is well-proven, and has wide acceptance and strong software support. Today, many EPC business process models are maintained with the help of ARIS Toolset of IDS-Sheer AG [15]. In addition, ARIS Toolset is also able to support export and import of business process models in a proprietary XML-based interchange format, AML (ARIS Markup Language) [10, 16], as well as multiple-users. These features of ARIS Toolset greatly facilitate the development of a supporting software tool for the proposed methodology.

In our approach, ARIS is used directly to generate OWL-S specifications. A possible alternative would be to use ARIS to generate OWL-S specifications indirectly via WSDL [2, 13, 16]. However, due to the inabilities of WSDL to adequately describe control flows, the alternative approach has severe limitations and is therefore not adopted.

3. ARIS Architecture

ARIS is comprehensive in the sense that it provides a "complete" description of a business process from five different views, namely organisation, function, product/service, data, and process/control. The organisation view describes the entities involved in the business process, along with their communication relationships. The function view depicts the activities to be executed, as well as their sequence. The product/service view describes concrete products or unsubstantial services and their flows, including fund flows. The data view shows the information objects of the business process and the data interchanged among them. Finally, all the views are integrated within the process/control view by a consolidated ARIS business process model. In addition, each of the five views is described at three different levels of details required to support the three phases of business process development, namely requirement definition, design specification, and implementation description. Figure 1 below shows the architectural framework of ARIS – the ARIS House [14].

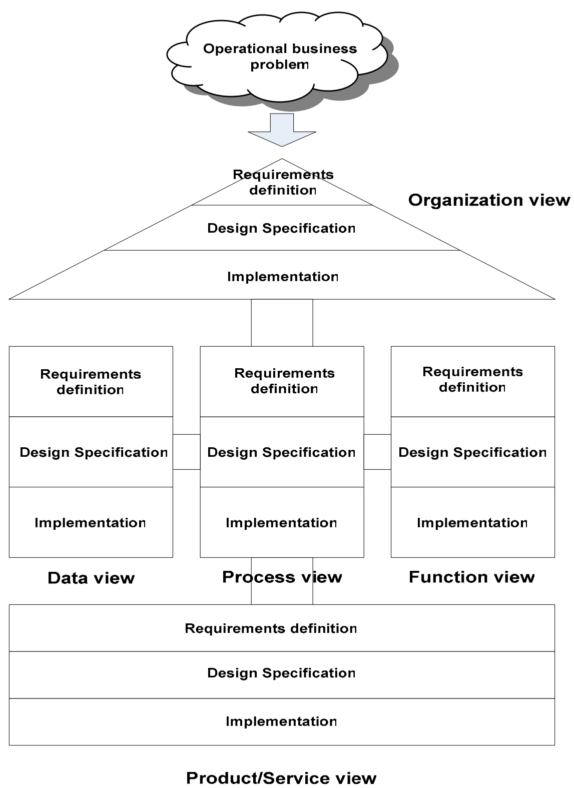

Figure 1: ARIS Architecture – The ARIS House

4. OWL-S

OWL-S is a SWS description language that enriches Web service descriptions with semantic information from OWL ontologies [12]. It defines three basic elements of a Web service: (1) a Service Profile that describes the capabilities of the service along with additional features that help to define the service, (2) a Process Model that provides a description of the activities of the service and the requester-provider interaction protocols, and (3) a Service Grounding that is a description of how abstract information exchanges described in the Process Model are mapped onto actual messages that the provider and the requester exchange.

A Web service capability describes what the service does in terms of the transformation it produces. The Process Model shows how the service does. Finally, the Service Grounding shows how to access the service. A Process Model is defined as an ordered collection of processes, where each process produces a state transformation. There are two types of processes: atomic processes and composite processes. Atomic processes correspond to operations that the provider can perform directly. Composite processes are collections of processes, atomic or composite, organised on the basis of some control flow structure like sequences, loops, concurrencies, etc.

The Service Grounding specifies how the abstract descriptions of information exchanges of atomic processes are transformed into concrete messages or remote procedure calls over the Internet.

5. Mapping between ARIS and OWL-S

As explained above, the ARIS architecture offers five different views of a business process with each view being described at three different levels of details. For the purpose of mapping between ARIS and OWL-S, the process/control view should be used. Being an integrated view, it offers in an integrated business process model a complete description of the business process. Our study [9] shows that, as far as functionality is concerned, requirement definitions of the process/control view are adequate to provide all the information that is necessary for the generation of the service profile and the process model specifications. The generation of the service grounding specification involves basically a one-to-one mapping from the atomic processes of the process model to the WSDL input and output message specifications. Tables 1, 2 and 3 show the ARIS notations of the OWL-S concepts as defined in OWL-S 1.0 [12].

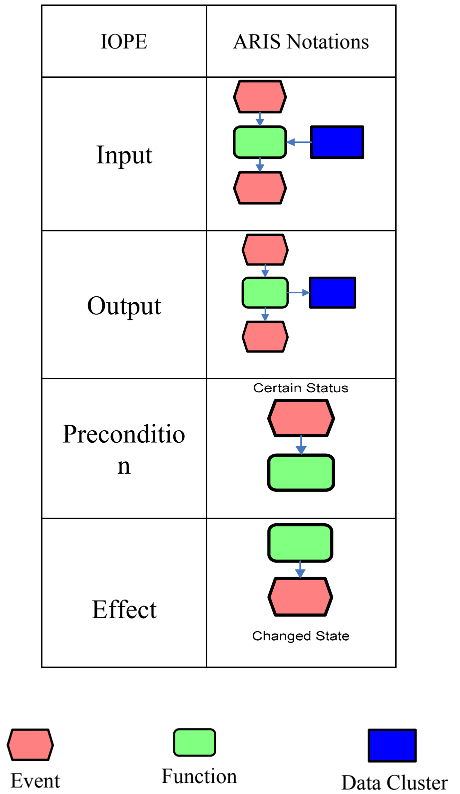

Table 1a: ARIS notations of OWL-S's IOPE

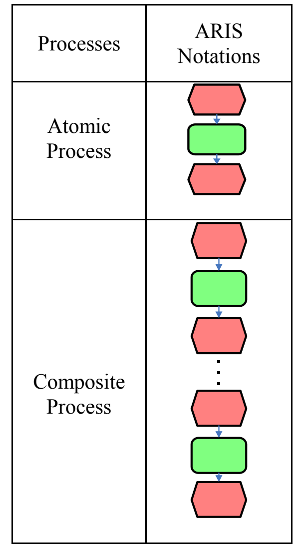

Table 1b: ARIS notations of OWL-S's Atomic and Composite Processes

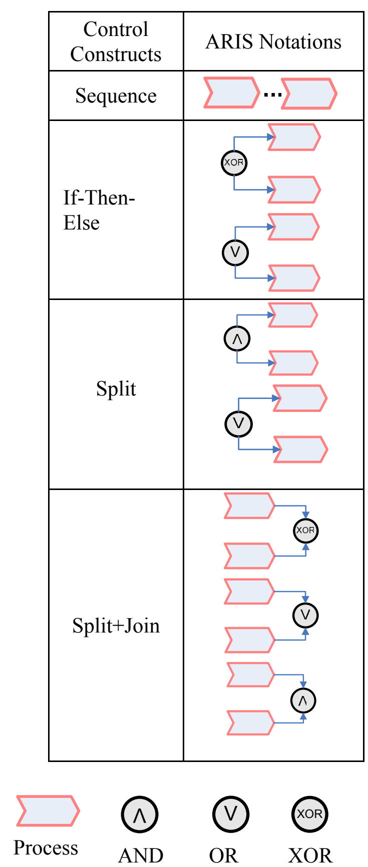

Table 1c: ARIS Notations of OWL-S Control Constructs

Design specifications of the process/control view are redundant as they are basically the standard Web service design specifications that are common for all Web service applications. So are implementation descriptions of the process/control view as they are mainly technical specifications which should be input, where necessary, manually to the OWL-S specifications by the service provider. Examples of such inputs include access methods and URLs of the service providers. It serves very little or no purpose automating the task of inputting such information. Mapping between ARIS and OWL-S is therefore confined practically to that between the requirement definition of ARIS's process/control view and the two OWL-S basic elements: Service Profile and Process Model (Figure 2).

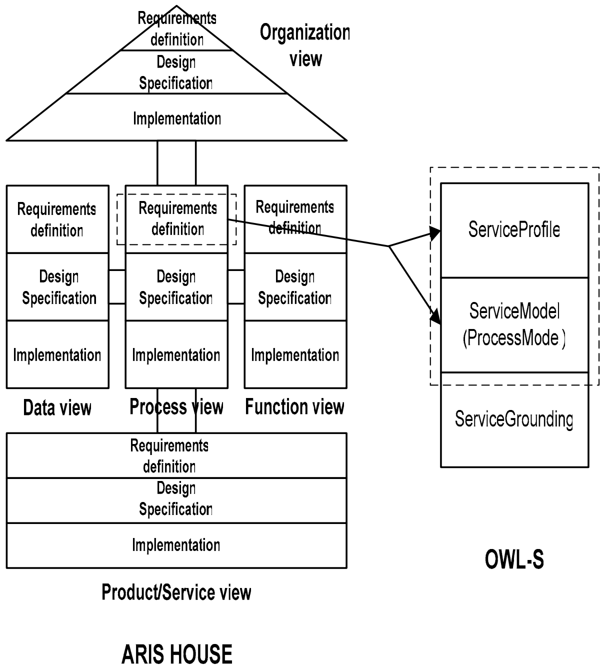

Figure 2: Mapping between ARIS and OWL-S

6. Methodology

The proposed methodology is described below from the perspectives of a service requester, a service provider, and a system administrator. For the service requester, the methodology describes how ARIS models are used for service discovery and service composition. For the service provider, it describes how the models are used for service publication. Finally, for the system administrator, it describes how to provide semantic descriptions of the service elements abstracted from the models.

6.1 Service Discovery

The IDEF0 diagram in Figure 3 describes how ARIS models are used for service discovery. The requester first creates in ARIS a service profile of the required service. The profile is then mapped onto OWL-S to create an OWL-S service profile which in turn is annotated with respect to the domain ontology in OWL for the semantic descriptions of the service elements abstracted from the ARIS model. The result is an OWL-S service profile specification providing a semantic description of the capabilities of the required service.

The same service profile in ARIS is also used to generate queries to semantic service registries for identifying eligible services. As OWL-S does not currently handle queries, OWL-QL [6] may have to be used for the purpose. Service discovery consists of matching the required capabilities with the available capabilities to identify eligible services. If more than one eligible service is identified, the best service is then selected based on cost, service level or some other non-functionality criteria. As the requester initiated his query without knowing beforehand which service he is to select and the inputs or preconditions required for the service to generate the outputs, the system must have the ability to query additional information from the requester.

The OWL-S service profile specification of the selected service is retrieved from the service registry. If necessary, the OWL-S specification is converted back to ARIS for the requester's perusal, by a reverse engineering process.

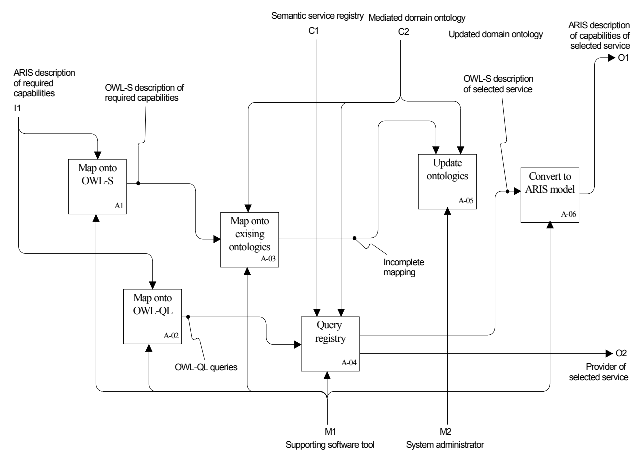

Figure 3: IDEF0 diagram for service discovery

6.2 Service Composition

To compose a Web service, the requester first creates an ARIS service profile of the required service and uses it to generate an OWL-S service profile specification of the required service as mentioned in 6.1. The only difference here is that the OWL-S service profile specification is used instead to query the semantic service registries for a collection of available services that can add up to the required service, i.e. an eligible service composition. If more than one eligible service composition is identified, the one that best matches the required service in terms of some non-functionality criteria is selected. The result is an OWL-S process model specification of the selected composite service showing the constituting services and their relationships.

6.3 Service Publication

For service publication, the provider must supply a complete OWL-S specification to a service registry of the service he is to provide. To do that using ARIS, he first has to create in ARIS a service profile, as well as a process model of the service. He then transforms them into the corresponding OWL-S specifications in the same way the service requester does in sections 6.1 and 6.2. The OWL-S service grounding specification is then obtained by mapping the atomic processes in the OWL-S process model specification directly onto WSDL [13].

Additional information that is needed for the generation of the complete OWL-S specification may be input manually by the service provider. Such information may include textual descriptions and contact information of the service provider, as well as technical information like access methods and URLs of the service provider.

The IDEF0 diagram in Figure 4 depicts how ARIS models are used for service publication.

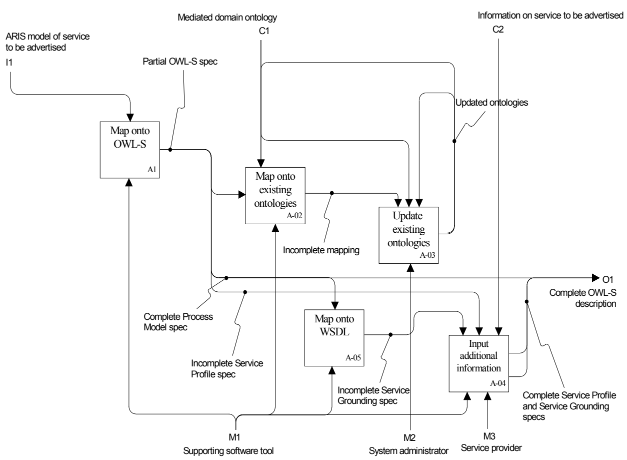

Figure 5: IDEF0 diagram for Web service publication

6.4 Semantic Description of Service Elements

The semantics of service elements abstracted from ARIS models can be described by annotating them with respect to the domain ontology. For the system administrator, his main task is to maintain and update the domain ontology. Though the requester and the provider may use the same ontology language for describing service elements, which is OWL in our case, they still may not share the same common semantics, hence the need for mediation. This requires constant updating of the domain ontology by the system administrator to accommodate all possible variations in terminology and definitions the users may introduce from time to time.

7. Framework

Figure 6 shows the overall process workflow for the transformation system. In this framework, the user, i.e. a service requester or provider, is responsible for creating ARIS models. An ARIS-OWL-S bi-directional converter will automatically convert the ARIS models to OWL-S specifications which are then used by the matchmaker for the selection of eligible services, or are sent to a semantic service registry for publishing. The same OWL-S specifications are also used by the service composer to identify eligible service compositions in the case of composite services. The converter can also automatically convert OWL-S specifications of the selected services back to ARIS models for the requester's perusal. The evaluator evaluates eligible services or service compositions against some non-functionality criteria to identify the one that best meets user requirements. Finally, semantics differences are resolved by an ontology mediator, with or without interventions of the system administrator. The framework therefore manages the transformation process with very little additional effort on the part of the user.

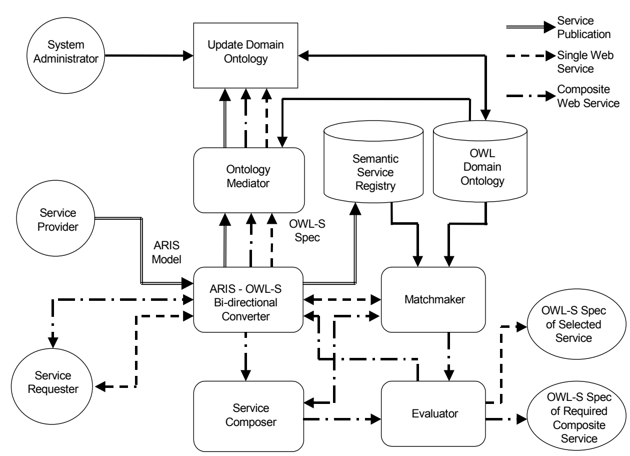

Figure 6: Framework of the Transformation Approach

8. Research Issues

The following are research issues that have yet to be resolved for the development of the ARIS-based transformation system.

8.1 Automatic Abstraction of OWL-S Specifications from ARIS Models

To automate the abstraction of OWL-S specifications from ARIS models, mapping between OWL-S and ARIS must be defined. A complete mapping from ARIS to OWL-S is not necessary: some of the mappings are redundant. So the mapping scope must first be defined to avoid unnecessary work. Mapping within the scope is then defined, and is realised in the form of an algorithm and a program code for the automatic transformation from ARIS to OWL-S. The development of the algorithm is by no means trivial as the two languages are very different: one is meant for human experts and the other for computers. The mapping here is never a simple one-to-one [9]. In addition, the service elements abstracted from the ARIS model must also be annotated with respect to the OWL domain ontology for semantic descriptions.

8.2 Automatic Service Composition

For automatic service composition, the system must have the capability to automatically identify eligible service compositions, and to automatically evaluate the compositions against non-functionality criteria like cost, time, service quality, etc., in addition to functionality criteria. This requires that semantic descriptions for the service level elements of Web Services be provided. Then, as far as the proposed transformation system is concerned, we are faced with the following research challenges:

1) How to enhance/enrich ARIS and OWL-S so that they can be used to capture service level requirements,
2) How to automatically map these requirements from ARIS onto OWL-S, and
3) How to develop a service ontology for the semantic description of the service level elements abstracted from enhanced/enriched ARIS.

Besides, a technique that can identify a collection of correlated services where their interim outputs can satisfy each other's input requirements and where the final deliverable meets the user requirements, must be developed.

For the evaluation of eligible service compositions, there are two possible approaches. The first assumes that a service composition can be defined at design time but may or may not be adaptively recomposed at run-time. The second assumes that a service composition may not be defined at design time but can be assembled service by service at run-tine. Both approaches have been well investigated in job shop studies over the last several decades, using various techniques ranging from heuristic rules, OR through AI to computer simulation. Research results and findings in these studies may be used here for the development of the transformation system.

8.3 Reverse Generation of ARIS Models from OWL-S Specifications

It is desirable to convert the OWL-S description of the selected service composition back to ARIS for the requester's perusal before invocating the service, especially if the process involved is complex and the consequence of a mismatch grave. The reverse generation process is much more complicated than the forward process mentioned in section 8.1 of generating OWL-S specifications from the corresponding ARIS models. Here we not only need to consider the control flow, but also how to represent the flow graphically and automatically in ARIS, whereas graphical information is ignored in the forward process.

8.4 Service Interactions

The proposed transformation system is based on current SWS techniques that take a simplistic view of service invocation. They do not accommodate ongoing interactions between service requesters and providers, nor do they support descriptions of legal protocols of interactions among them. Currently SWS techniques are limited to services that are "closed." They are therefore inadequate, for example, in complex B2B or e-commerce situations where services are open involving ongoing interactions with each other and with service requesters. With the current SWS techniques, it is possible to invocate services that are less than desired. For example, if an e-commerce service does not allow orders to be changed after a certain time has elapsed, it may be unsafe to use this service in certain settings, even if it is superior in other respects. Therefore we are faced with three main challenges here:

1) How to enhance/enrich ARIS and OWL-S so that they can also be used to specify service interaction requirements,
2) How to automatically map interaction requirements from enhanced/ enriched ARIS onto enhanced/enriched OWL-S, and
3) How to develop a communication ontology for the semantic description of the communication elements abstracted from enhanced/enriched ARIS.

These are problems which must be resolved for the ARIS-based transformation system to be useful in real life.

8.5 Systematic Domain Ontology Building

The proposed transformation system is ontology-based. For such a system, a genuine methodology and software tool for domain ontology building is of vital importance. In a comprehensive survey of ontology-based systems, Wache H et al [17] concluded that there is a striking lack of sophisticated methodologies for the development of domain ontologies: most ontology-based systems either assume ontologies a priori or provide only partial methodologies.

There are basically two approaches to domain ontology building, federated and mediated. Most, if not all, industries still do not have a common shared ontology in use to date. Due to the difficulty of building a consensus about what terminologies and structures should be used, it is doubtful if such an ontology will ever exist. The mediated approach, in which each company has its own ontology and translations between different ontologies are by means of mediation, is more practical. It is this latter approach that the transformation system should be based on.

The mediated ontology approach imposes three specific requirements on the methodology:

1) The methodology must be able to support the systematic integration through mediation of a large number of existing ontologies,
2) It must be able to support incremental building of ontologies and handle ontology evolution, and
3) It must be based not just on theoretical analysis, but also on empirical analysis in order to be able to integrate local ontologies in the real-world.

In addition, it must of course be comprehensive, i.e., able to handle all types of sophisticated mappings between ontologies and ontology constraints.

The development of a systematic methodology will be a major step in the work on using ontologies for automatic specification and development of SWSs. It will help to integrate whatever techniques that have been developed or are being developed in various areas of ontologies and to put these techniques to work in real-life applications.

As far as we know, none of the existing methodologies has the sophistication to meet all these requirements. This is supported by the finding of a survey on ontology-based systems conducted by Wache et al [17] that there is a striking lack of sophisticated methodologies for the development of domain ontologies: most ontology-based systems either assume ontologies a priori or provide only partial methodologies.

9. Conclusion

Studies have shown that UML can be used to generate OWL-S specifications for SWS development [1, 5, 7, 8 and 17]. Since ARIS and UML are completely interchangeable as far as content is concerned [14], there is every reason to believe that ARIS can also be used for the same purpose. Our preliminary investigation as described in this paper supports the belief. In order to overcome the problem of OWL-S and to help realise the widespread use of SWSs, we have proposed an ARIS-based transformation methodology for the automatic specification and development of SWSs. We have also identified research issues that have yet to be resolved for the development of a supporting software tool.

References

[1] P Bose et al, Applying UML to Model Web Service Ontologies for the Semantic Web, Scientific American, pp 34-43, 284(5), 2001.

[2] E Christensen et al, Web Services Description Language, Version 1.1, 15 March 2001, http://www.w3.org/TR/2001/NOTE-wsdl-20010315.

[3] D Dori, Object-Process Methodology – A Holistic Systems Paradigm, Springer Verlag 2002.

[4] D Dori et al, Modelling Semantic Web Services with OPM/S – A Human and Machine-Interpretable Language, www.technion.ac.il/~erant/papers/Dori_OPM-S.pdf.

[5] K Falkovych et al, UML for the Semantic Web: Transformation-based Approaches, 2002, http://www.cwi.nl/~media/publications/UML_for_SW.pdf.

[6] R Fikes et al, A Language for Deductive Query Answering on Semantic Web, http://classweb.gmu.edu/kersch/infs770/Topics/SemanticWeb/KSL-03-14 (OWL-QL).pdf

[7] D Frankel et al, The Model Driven Semantic Web, http://www.sandsoft.com/edoc2004/FHKM-MDSWOverview.pdf.

[8] R Grønmo et al, Model-Driven Web Services Development, www.hpl.hp.com/techreports/2004/HPL-2004-107.html.

[9] Y Gu, An ARIS-based Model-Driven Approach to Semantic Web Service Development, MSc Thesis (partial fulfillment), in advanced preparation, School of EEE, NTU.

[10] J Mending, and M Nüttingens, Transformation of ARIS Markup Language to EPML, Proceedings of the 3rd GI Workshop on Event-Driven Process Chains (EPK 2004), Luxembourg, Luxembourg, October 2004.

[11] S Narayanan, and S McIlraith, Analysis and Simulation of Web Services, Computer Networks, April 2003.

[12] OWL Services Coalition, OWL-S: Semantic Markup for Web Services, Dec 2003, http://www.daml.org/services/owls/1.0/owl-s.html.

[13] M Paolucci et al, Towards a Semantic Choreography of Web Services: from WSDL to DAML-S, http://projects.semwebcentral.org/docman/view.php/30/5/wsdl2daml.

[14] A W Scheer, ARIS-Business Process Frameworks, page 44, Springer-Verlag Berlin-Heidelberg 1998.
[15] IDS-Sheer A G, ARIS Toolset for Business Reengineering, www.moonglow.com/ARIS.
[16] IDS-Scheer A G, XML Export and Import, ARIS Collaborative Suite Version 6.2, ftp://ftp.ids-scheer.de/pub/ARIS/HELPDESK/EXPORT/, Jun 2003.
[17] Wache H et al, Ontology-Based Integration of Information – A Survey of Existing Approaches, 2001 *http://www.tzi.de/buster/papers/SURVEY.pdf*