

# A Framework for Interoperability Analysis on the Semantic Web using Architecture Models

Johan Ullberg, Robert Lagerström, Mathias Ekstedt  
Department of Industrial Information and Control Systems  
Royal Institute of Technology (KTH)  
{johanu, robertl, mathiase}@ics.kth.se

## Abstract

*IT decision making requires analysis of possible future scenarios. The quality of the decisions can be enhanced by the use of architecture models that increase the understanding of the components of the system scenario. It is desirable that the created models support the needed analysis effectively since creation of architecture models often is a demanding and time consuming task. This paper suggests a framework for assessing interoperability on the systems communicating over the semantic web as well as a metamodel suitable for this assessment. Extended influence diagrams are used in the framework to capture the relations between various interoperability factors and enable aggregation of these into a holistic interoperability measure. The paper is concluded with an example using the framework and metamodel to create models and perform interoperability analysis.*

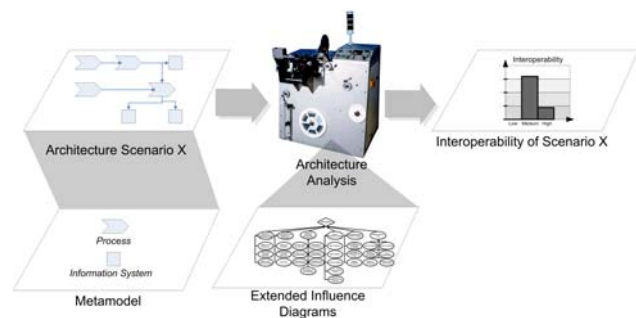
## 1. Introduction

The hopes on the semantic web as a solution to many of the problems with information systems interoperability are currently growing. Foremost are these hopes related to a future where information systems will start interacting in a more autonomous and intelligent way without humans first specifying the interaction in detail. However, the semantic web is not an unambiguous product that is ready to use off-the-shelf. Rather, the future users of the semantic web face a number of design decisions that they need to consider when integrating their information systems with this mechanism.

Architecture modeling is today a state of the art approach to information systems development and management. Essentially the main idea is that the architecture models should predict the behavior of the information system, acting in its environment, before the information system is developed and is being taken into operation. The architecture models allow reasoning about the consequences of various potential scenarios and thereby support decision making. Using models also

increases the understanding of complex systems and enables reuse of information.

In order to predict which architecture scenario is preferable, three things are needed. Firstly, models over the scenarios need to be created. Secondly, it is necessary to define what is desirable; the goal. In this article the goal is to achieve high information system service interoperability. Thirdly, we need to understand the causal chains from the scenario choice to the goal. Suppose that scenario A features services described in a semantic web language using an ontology that is very suitable for expressing the current service as such, but the language and the ontology is not widely spread. In scenario B on the other hand, the service is described in a semantic web language that is wide spread but has several unambiguous interpretations. To decide which scenario is preferable is often difficult, particularly without a formal analysis.



**Figure 1. The relation between metamodels, architecture scenarios, analysis, formal specification of analysis, and the result of the analysis**

In order to perform this kind of analysis, the architecture models firstly need to contain the proper information. In the above example, where the decision maker is interested in service interoperability over the semantic web, the models need to contain information regarding which languages and ontologies that are used for describing the service, how expressive those languages and ontologies are, how easy it is to identify

the service descriptions, etc. The kind of information contained in a model is given by its metamodel, so it is important that architecture metamodels are properly designed.

In order to determine if a metamodel is amenable to the analysis of a certain quality attribute, such as interoperability, it would be helpful with a structured account of that analysis. Which of all the modeled aspects are most important and which aspects are depending on each other and how? We will use a notation called Extended Influence Diagrams (EID) [1] in order to formalize the analysis of interoperability.

Figure 1 depicts the relation between an architecture scenario, modeled using a metamodel, the analysis of the scenario, the formal specification of the analysis through an extended influence diagram and finally the output: the interoperability level of the analyzed scenario.

The main contribution of this paper is an extended influence diagram and a metamodel that, through the creation of architecture models, supports interoperability assessments of information system services using the semantic web.

The remainder of this paper is delineated as follows; section 2 introduces some concepts of the semantic web. Extended influence diagrams are introduced in section 3. Section 4 presents the framework for semantic web service interoperability analysis in the form of an extended influence diagram. Section 5 evaluates the usefulness of a number of common architecture metamodels. Section 6 proceeds to detail the content of the metamodel that supports the interoperability analysis. The applicability of the metamodel is demonstrated in the subsequent section 7. Finally, section 8 concludes the paper.

## 2. Semantic Web Concepts

This section briefly presents the different concepts related to the semantic web. These concepts are later introduced as entities for the semantic web metamodel, see Figure 5.

The overall use case for the semantic web is that *Information Providers* publish information and the *Agents* task is to find the information that is correct with respect to the agents *Goal*. In other words the information provider and the agent does in the successful case perform what we here label *Stakeholder Collaboration*. The information provider publishes semantic information about *Things* in the real world in a *Formal Denotational Description*, i.e. the semantic part of a web webpage. This might e.g. be goods for sale or services provided. The agent has the goals encoded in a *Requirement Description*. We thus here differentiate the real-world objects and phenomena Things and Goals from those belonging to the technical semantic web solution. Both

the requirement description and the formal denotational description are written in a *Semantic Language* such as RDF [2].

Semantic descriptions use *Ontologies*, explicit specifications of a conceptualization [3], to define the meaning of terms in their descriptions. Well known ontologies available today include for instance Dublin Core [4] for documents, SUMO[5] as an upper ontology describing more abstract concepts and numerous domain specific ontologies such as for instance the Gene Ontology project [6]. These ontologies are also written in a semantic language, e.g. DAML+OIL [7] and OWL [8], where nowadays perhaps OWL is the most well known and endorsed by W3C. In order to relate different ontologies to each other *Ontology Gateways* are used for this mapping alignment or merging. [9][10]

In the search for relevant information the agent can use *Information Retrieval Applications*, i.e. search engines that crawls the web and index the content it find. Swoogle [11] is one example of such an application. The information retrieval applications will be queried using a *query language* such as Corese [36], RQL [12] or SquishQL [13].

## 3. Bayesian Networks and Extended Influence Diagrams

Friedman describes a Bayesian network,  $B=(G, P)$ , as a representation of a joint probability distribution, where  $G=(V, E)$  is a directed acyclic graph consisting of vertices,  $V$ , and edges,  $E$  [14]. The vertices denote a domain of random variables  $X_1, \dots, X_n$ , also denoted chance nodes. Each chance node,  $X_i$ , may take on a value  $x_i$  from the finite domain  $Val(X_i)$ . The edges denote causal dependencies between the nodes, i.e. how the nodes relate to each other. The second component,  $P$ , of the network  $B$ , describes a conditional probability distribution for each chance node,  $P(X_i)$ , given its parents  $Pa(X_i)$  in  $G$ . It is possible to write the joint probability distribution of the domain  $X_1, \dots, X_n$  using the chain rule of probability, in the product form:

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i))$$

In order to specify the joint distribution, the respective conditional probabilities that appear in the product form must be found. The second component  $P$  describes distributions for each possible value  $x_i$  of  $X_i$ , and  $pa(X_i)$  of  $Pa(X_i)$ , where  $pa(X_i)$  is the set of values of  $Pa(X_i)$ . These conditional probabilities are represented in matrices, here on called conditional probability matrices (CPMs).

If the probabilities of the source variables are known, it is possible to infer a value for the target variable using the law of total probability,

$$P(X_1) = \sum_i P(X_1 | Pa(X_i)) P(Pa(X_i))$$

Also, using Bayes' rule,

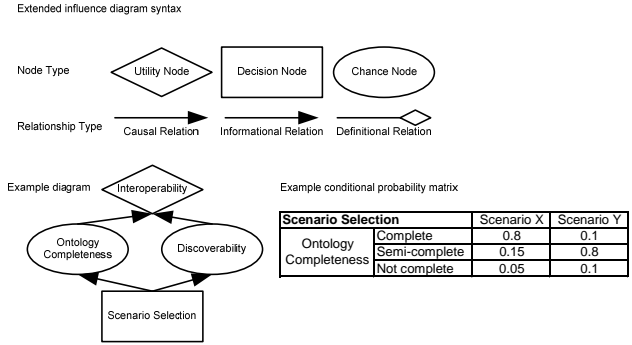
$$P(X_i | Pa(X_i)) = \frac{P(Pa(x_i) | X_i) P(X_i)}{P(Pa(x_i))}$$

makes it possible to calculate the values of source variables based on the probabilities of a target variables.

Extended influence diagrams are an extension of influence diagrams [15][16], which in turn are an enhancement of the above mentioned Bayesian networks [17][18]. Thus, extended influence diagrams support probabilistic inference in the same manner as Bayesian networks do; given the value of one node, the values of related nodes can be calculated. The different relations that can be used in an extended influence diagram are either causal, as in Bayesian networks, informational, or definitional. In extended influence diagrams there are three different types of nodes; decision nodes, utility nodes, and like in Bayesian networks chance nodes. Decision nodes represent the decisions that can be made, e.g. selecting between different architecture scenarios. Utility nodes represent the goals, e.g. semantic web interoperability. Chance nodes could typically be ontology completeness or discoverability. The syntax for the graphical representation different relations and nodes is presented in Figure 2.

An example extended influence diagram with a conditional probability matrix is presented in Figure 2 below the extended influence diagram syntax. In this example the probability matrix represents the probabilities of the attribute ontology completeness to be complete, semi-complete, or not complete if a scenario x or a Scenario Y is selected. As can be seen in the figure the ontology completeness would be semi-complete with a probability 0.15 (15 %) if scenario x is selected.

For more comprehensive treatments on influence diagrams and extended influence diagrams see [1], [15], [16], [17], [18], [19], [20] and [21]. The extended influence diagrams here serve the purpose of formally specifying architecture analyses.



**Figure 2. An extended influence diagram and a simple example. With a chosen scenario in the decision node, the chance nodes will assume different values, thereby influencing the utility node [22].**

#### 4. A Framework for Interoperability Analysis on the Semantic Web

This section presents an extended influence diagram that captures theory regarding interoperability from the field of the semantic web. The design of the extended influence diagram is mainly influenced by [23][24][25][26][27][28][29][30].

Several methods for assessing interoperability on a general scope have previously been suggested. The assessment methods include LISI[31], SoSI[32], LCIM[33] and i-Score[34]. Few of these are focused on applying mathematical models and not all have a numerical measurement of interoperability [34]. The work presented in this paper incorporates many of the proposed interoperability measures and uses the mathematics of Bayesian networks as a means of aggregation.

Interoperability is the ability of two or more systems or components to exchange information and to use that information [35]. Adopted to the domain of the semantic web, *Semantic Web Interoperability* is defined as the probability for successful retrieval of information on the semantic web. Semantic web interoperability is influenced by five concepts: firstly, *Transmission protocol compatibility*, meaning that the transmission protocols of the agent and the information provider are compatible (generally http); secondly, *Discoverability*, meaning how difficult it is to find the appropriate information provider; thirdly, *Ontology completeness*, i.e. that the Ontologies have enough coverage and expressiveness; fourthly, *Quality of formal denotational description markup*, concerned with the markup of the provided information that is to be found; finally, *Quality of requirement description markup*, meaning that the

specification of what is sought after also is marked up in a sufficient way, cf. Figure 3.

#### 4.1 Discoverability

Discoverability in the semantic web is mainly concerned with two tasks [11]: firstly, finding appropriate ontologies while performing markup, *Ontology discoverability*, and secondly, finding the instance data on pages containing semantic markup, captured by the concept *Quality of information retrieval application*.

The quality of the information retrieval application is dependent on its ability to perform indexing, *Quality of indexing*, and the *Semantic expressiveness of query language*. Query languages such as Corese[36], RQL[12] and SquishQL[13] have different expressiveness in terms of formally specifying the sought after information.

#### 4.2 Ontology Completeness

The *Ontology completeness* is concerned with coverage of the ontologies, that they cover all aspects needed to create a rich formal semantic description of the objects in the real world as well as the objectives for

someone searching for information. The general ontology completeness is defined in terms of the completeness of the ontologies related to the requirement description, the formal denotational description and possible ontology gateways respectively.

The *Requirement description's ontology completeness with respect to real world goal* is the matter of ensuring that the ontology that the user, seeking information, applies when creating a requirement description for the agent is complete with respect to the goal the user has. In the same manner someone publishing information relate to one or many ontologies that also have to be complete with respect to the thing in the real world that is to be described. This concept is named *Formal denotational description's completeness with respect to real-world entity* in the extended influence diagram. Generally, the ontologies of the requirement description do not match those used in the formal denotational description. Ontology gateways can mitigate this problem by relating ontologies to each other. It is however of importance that the gateways are complete with respect to the requirement description, *Ontology gateway completeness with respect to requirement description*.

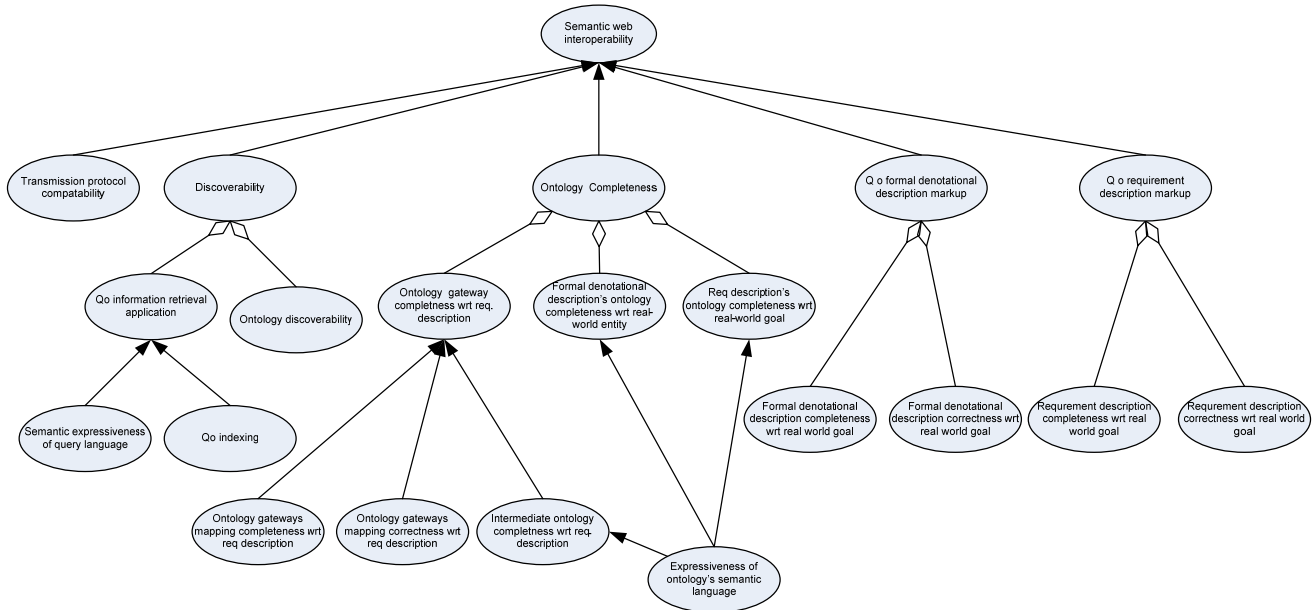


Figure 3. The extended influence diagram containing factors influencing semantic web interoperability and thereby of interest when performing analysis of such

The *Expressiveness of ontology's semantic language* influences the ontology completeness, so that all wanted relations between concepts can be expressed in the language. The ontology gateway completeness is also dependent on: the *Ontology gateway's mapping completeness with respect to requirement description*, i.e. that all concepts in the requirement description are mapped in the gateway; the *Ontology gateway's mapping correctness with respect to requirement description*, i.e. that all mapped concepts are correctly mapped; and finally the *Intermediary ontology completeness with respect to requirement description*. Sometimes the ontologies aren't mapped directly to each other but rather through one or more intermediary ontologies. If so it's of importance that these ontologies are complete in coverage of the important concepts.

### 4.3 Quality of Formal Denotational Description Markup

The basis for the semantic web is that information on the web is marked up in a formal denotational description. The quality of this markup is thus of importance to the interoperability. This quality is defined in terms of *Formal denotational description completeness with respect to real world goal*, the all (relevant) information is semantically marked up, and *Formal denotational description correctness with respect to real world goal*, that the markup is correctly performed.

### 4.4 Quality of Requirement Specification Markup

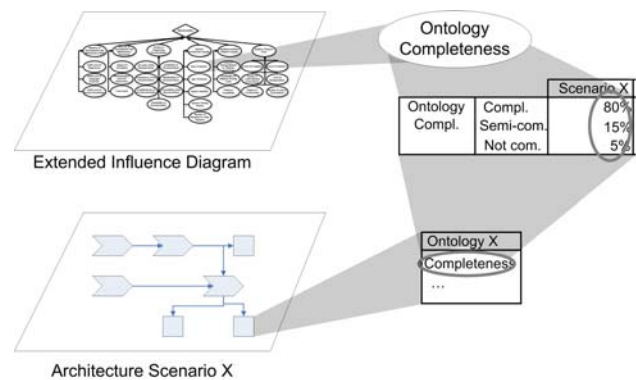
The requirement description, the coding of the goal the agent tries to fulfill, should also be marked up semantically in order to achieve interoperability. As in the case with the formal denotational description this quality is defined as *requirement description completeness with respect to real world goal*, the all information is semantically marked up, and *requirement description correctness with respect to real world goal*, that the markup is correctly performed.

## 5. Architecture Frameworks for Analysis

With the requirement on architecture models to support architecture analysis follows a specific requirement on architecture metamodels. Specifically, all entities and attributes that are required for a complete analysis as specified in an extended influence diagram must be found in the architecture metamodel, in order for the corresponding model to be amenable to analysis. See Figure 4.

There exist many architecture modeling frameworks and languages. The number one software system

modeling language is UML [37]. UML provides a metamodel divided into a number of diagram types that can be used for system design and analysis. There also exist extensions to UML such as SysML [38] which adopts hardware aspects of systems to the models. There also exist a substantial number of enterprise architecture frameworks that also takes business and the usage of systems into account in the models. Two enterprise architecture frameworks that are explicitly focused on metamodels are the Department of Defense Architecture Framework (DoDAF) [39] and Archimate [40] Finally there the framework specifically for interoperability such as ATHENA Interoperability Framework (AIF) [41], Levels of Systems Interoperability (LISI) [31] and the European Interoperability Framework (EIF) [41].



**Figure 4. The properties found in an extended influence diagram determine what entities and attributes should be present in an architecture metamodel.**

When considering the suitability of the metamodels related to these frameworks to the architecture analysis considered in preceding sections, we have found significant difficulties. The metamodels are not detailed enough to provide the information required for the analysis. We are interested in information such as for instance the ontologies how easy it is to identify them. However, the concept of "ontology" is not found in the metamodels. In addition we are interested in a specific attribute of the ontology and some metamodels do not even systematically propose attributes at all.

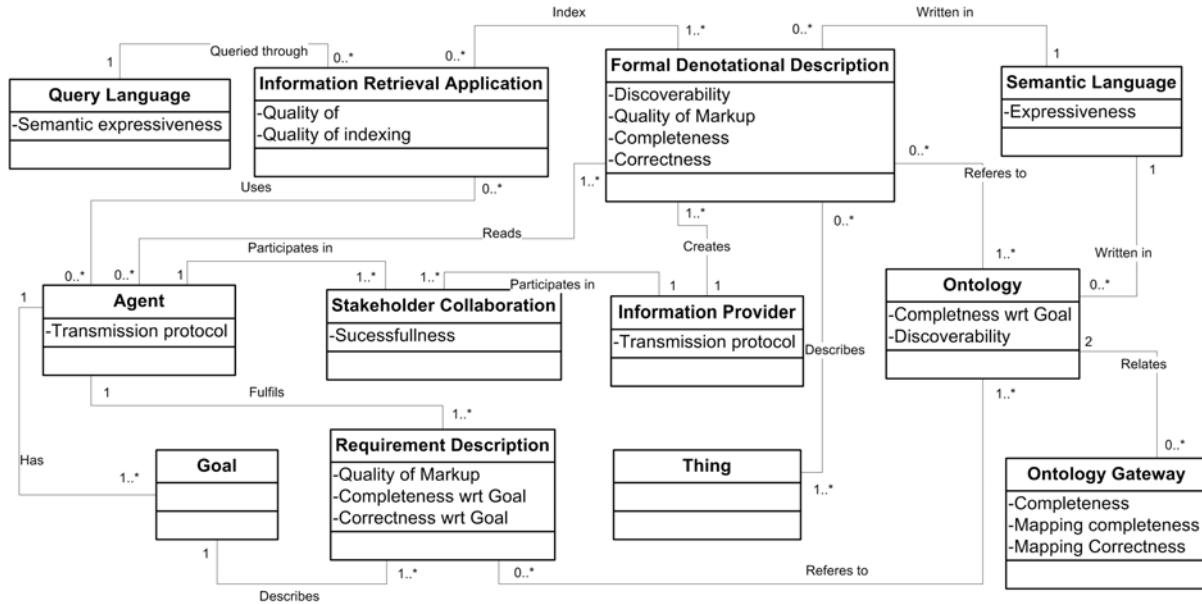


Figure 5. The metamodel for semantic web interoperability analysis with its entities, attributes, and relations.

## 6. The Metamodel for Interoperability Analysis on the Semantic Web

In this section, the metamodel suggested for semantic web interoperability analysis is presented. For the sake of interoperability analysis, only the extended influence diagram of section 4 is needed, but as discussed in the introduction, models can be useful in the decision making process. The metamodel is constructed to satisfy the requirements of the preceding section, i.e. it contains all of the entities and attributes necessary to conduct analysis of interoperability.

### 6.1 Entities of the Metamodel

The entities of the metamodel presented in Figure 5 have all been introduced in section 2 as concepts of the semantic web and will here only be listed with their relationships; *Information Providers* create *Formal Denotational Descriptions* that describes *Things* in the real world. The *Agent* performs a *Stakeholder Collaboration* with the information provider and fulfils a *Requirement Description* which describes a *Goal*. To be able to do this the agent must read formal denotational descriptions and understand its' content. Both the requirement description and the formal denotational description are written in a *Semantic Language* and relate to an *Ontology*. Ontologies can be related to other ontologies using *Ontology Gateways*. The agent can use *Information Retrieval Applications* as an aid. These applications are queried through a *Query Language*

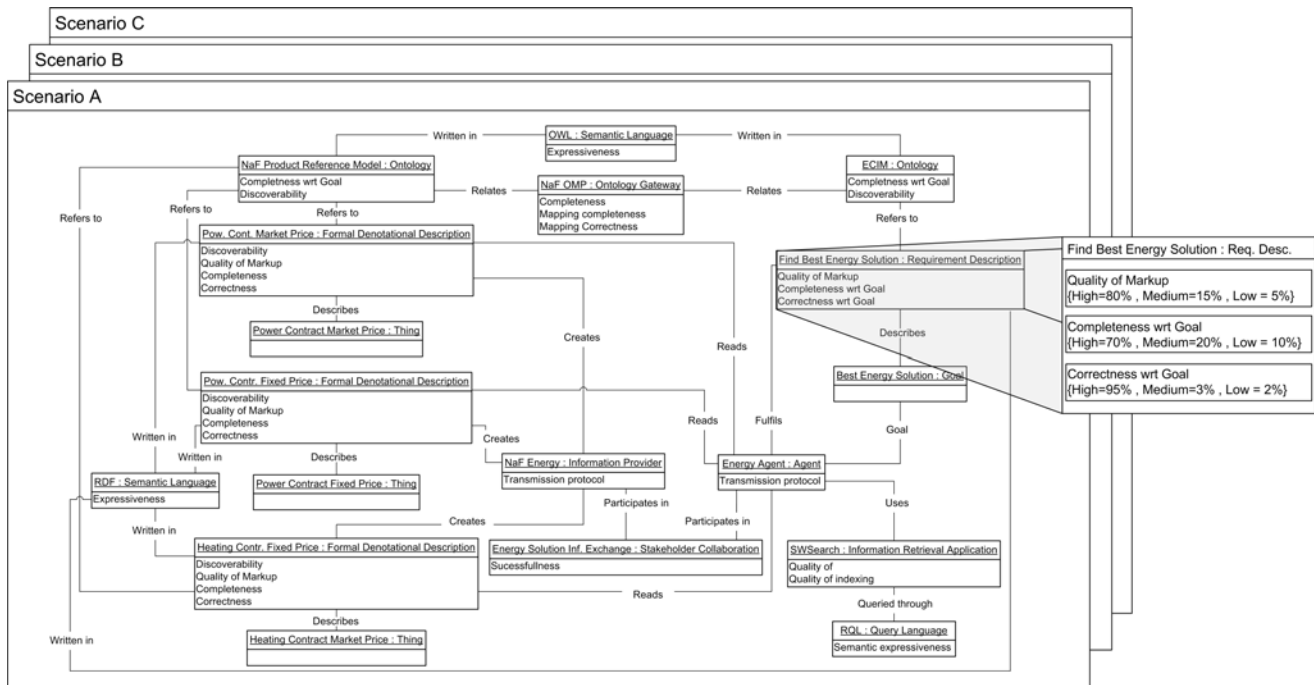
### 6.2 Attributes of the Metamodel

For the purpose of semantic web interoperability analysis, a metamodel without attributes would be inadequate. In an architecture model, many important concepts are best captured as entity attributes. As seen in Figure 5, some entities have attributes that correspond to the extended influence diagram of section 4.

Firstly the sought after attribute interoperability, defined as successful collaboration on the semantic web, can be found as the attribute *successfulness* in the entity stakeholder collaboration. The formal denotational description contains four attributes, *discoverability* matching the node with the same name in the extended influence diagram, *quality of markup* and the two attributes affecting this quality, *completeness* and *correctness*. In similar manner the requirement description also contains the attributes *quality of markup*, *completeness* and *correctness*.

In order to match the extended influence diagram the other attributes of the metamodel are the *discoverability* and *completeness* of the ontology, the *expressiveness* of the semantic language, the ontology gateways *mapping completeness* and *correctness* and the *semantic expressiveness* of the query language.

There is one variable in the extended influence diagram not directly related to one attribute in the metamodel, namely the transmission protocol compatibility, this is evaluated by a comparison of the attributes *transmission protocol* in the entities agent and information provider.



**Figure 6. The architecture model of scenario A, one out of three scenarios in the example. The requirement description entity is magnified to show the probability distributions related to each attribute.**

## 7 Modeling and Analyzing Using the Metamodel – An Example

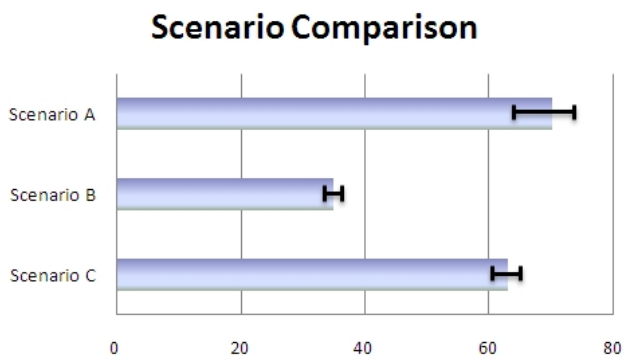
This section presents an example of a semantic web interoperability analysis used as decision support in a project at the power company NaF Energy. The management of NaF Energy wants to increase the number of customers and therefore marketing campaigns have been initiated. As a part of this campaign a project has been started in order to improve the services offered by NaF and how these services are published on the web. The chief architect in the project suggested that the product portfolio should be published on the semantic web. The product portfolio consists of three offerings; one power contract at market price, one power contract at fixed price, and one heating contract at market price. There are several architectural options for our architect to consider in order to achieve the goal, i.e. that agents will find information regarding the services offered by NaF. There is for instance the choice of which ontology to use, some are more suitable for describing NaF's product portfolio, others are less appropriate but more widely used by the public. NaF can also choose on how much effort, e.g. time, to put into the project of marking the description of their products.

Several possible scenarios are therefore considered and the architect decides that a formal evaluation of the candidate scenarios is to be performed. Based on the metamodel of Section 6 information on the entities and their attributes are collected. Figure 6 describes one scenario, scenario A, in which an in-house and therefore very well suited but not well known ontology is used as well as an ontology gateway for the mapping towards more established ontologies.

Information collection can be done in several ways. In this example the expressiveness of the languages used was assessed by an expert. The completeness of the ontologies was found by looking in the actual ontology and comparing it to what was to be described. The completeness of the mappings towards the ontologies was assessed by interviewing the developers that should implement the solution and asking them how complete the mapping would be given the budget.

All collected variable values were then translated into discrete states, such as Low, Medium, or High. These were then used as input to the semantic web interoperability analysis employing the Bayesian theory in extended influence diagrams, as described in Section 3. When collecting information for the models, there is an issue of credibility [42]. Low credibility may lead to a large uncertainty in the analysis, making it difficult for

the architect to make a rational decision. For instance, studying the actual ontologies to find their completeness is a tedious work but, if done well, this will provide the architect with high credibility of the gathered information. Whereas, interviewing personnel, e.g. developers, to find the completeness in the mapping is less credible and also dependent on the experience of the personnel and the bias of the interviewer. Oftentimes it is very expensive to collect the information needed for a perfectly credible analysis. Since the analysis is based on the formalism of extended influence diagrams this credibility variation can be handled, thus the presented method of analysis provides the architect with an uncertainty degree in the result, shown in Figure 7 as bars indicating the range of values the result may assume.



**Figure 7. The comparison between the service interoperability of the different scenarios, the black I-bars indicate the uncertainty of the assessments.**

The final result of the analysis is shown in Figure 7. As can be seen from the figure, scenario A achieved the highest interoperability rating whereas scenario B have a considerably lower degree of interoperability due to the use of an ontology not well known and not mapped to other, more well known, ontologies. Even though not detailed in this paper, using extended influence diagrams for analysis allows for assessments of subcomponents and it is therefore possible to discover that scenario A achieves its' high interoperability score due to a high degree of mark-up of both the requirement description and the formal denotational description, while the ontology completeness scores slightly lower due to incorrectness of the ontology mappings. Scenario C on the other hand has a lower degree of mark-up and discoverability but a higher degree of ontology completeness. The architect can now, based on the interoperability score, make a rational decision, choosing an architecture providing the degree of interoperability needed by the enterprise.

## 8 Conclusion

This paper has presented a framework for interoperability analysis on the semantic web and a metamodel supporting the analysis. The metamodel consists of entities with accompanying attributes that can be used to create architecture models from which it is possible to extract precisely the information that is needed for quantitative semantic web interoperability analysis. An example was provided illustrating the use of the metamodel and the extended influence diagram for analysis.

## 9. References

- [1] Johnson, P., et al., "Enterprise Architecture Analysis with Extended Influence Diagrams" *Information System Frontiers*, vol 9(2), Springer, The Netherlands, 2007.
- [2] S. Decker, P. Mitra and S. Melnik, "Framework for the Semantic Web: An RDF Tutorial," *IEEE Internet Computing*, 2000, pp. 68-73
- [3] T.R. Gruber, "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition*, 1993, pp. 199-220.
- [4] Dublin Core, Dublin Core Metadata Initiative, available at <http://dublincore.org/>, 2005
- [5] I. Niles and A. Pease, "Toward a Standard Upper Ontology", *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, 2001.
- [6] B. Smith, J. Williams, and S. Schulze-Kremer. "The Ontology of the Gene Ontology" *Proceedings of AMIA Symposium*, 2003
- [7] A. Gomez-Perez and O. Corcho, "Ontology Languages for the Semantic Web", *IEEE Intelligent Systems*, 2002, pp. 54-60.
- [8] I Horrocks, P.F. Patel-Schneider, and F. van Harmelen. "From SHIQ and RDF to OWL: the making of a web ontology language" *Journal of Web Semantics*, 2003, pp 7-26.
- [9] Jurafsky, D., Martin, J.H., *Speech and Language Processing*, Prentice Hall, New Jersey, 2000.
- [10] Cardoso, J., Sheth, A., "Semantic E-Workflow Composition", *Journal of Intelligent Information Systems*, 21:3, Kluwer, The Netherlands, 2003, pp 191-225.
- [11] L. Ding et. al. "Swoogle: A Search and Metadata Engine for the semantic web", *Proceeding of the thirteenth ACM international conference on Information and knowledge management (CIKM)*, Washington, DC, 2004, pp 652-659
- [12] G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, M. Scholl, "RQL: a declarative query language for RDF", *Proceedings of the 11th international conference on World Wide Web*, Honolulu, Hawaii, USA, 2002, pp. 592-603
- [13] L. Miller, A. Seaborne and A. Reggiori, "Three Implementations of SquishQL, a Simple RDF Query Language," *Proceedings of 1st International Semantic Web Conf. (ISWC 2002)*, Springer-Verlag, 2002, pp. 423-435.
- [14] N. Friedman, M. Linial, I Nachman and D. Pe'er, "Using Bayesian Networks to Analyze Expression Data." *Journal of Computational Biology*, Mary Ann Liebert, Inc., publishers., 2000, pp. 601-620.

- [15] Shachter, R., "Evaluating influence diagrams" *Operations Research*, 34(6) Institute for Operations Research and the Management Sciences, Hanover Maryland, 1986, pp. 871-882.
- [16] Howard, R.A., Matheson, J.E., "Influence Diagrams" *Decision Analysis Vol. 2(3)*, Institute for Operations Research and the Management Sciences, Hanover Maryland, 2005, pp. 127-143.
- [17] Neapolitan, R., *Learning Bayesian Networks*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 2003.
- [18] Jensen, F.V., *Bayesian Networks and Decision Graphs*, Springer New York, Secaucus, NJ, USA, 2001.
- [19] Johnson, P., Lagerström, R., Närman, P., "Extended Influence Diagram Generation" *Enterprise Interoperability II – New Challenges and Approaches*, Springer, London, 2007, pp. 599-602.
- [20] Shachter, R., "Probabilistic inference and influence diagrams" *Operations Research*, 36(4), 1998, pp 36-40.
- [21] Johnson, P., Ekstedt, M.: *Enterprise Architecture – Models and Analyses for Information System Decision Making*. Studentlitteratur, Lund, Sweden, 2007.
- [22] Lagerström, R., "Analyzing System Maintainability Using Enterprise Architecture Models" *Proceedings of the 2nd Workshop on Trends in Enterprise Architecture Research (TEAR'07)*, St Gallen, Switzerland, 2007, pp. 31-39,
- [23] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web" *Scientific American*, Scientific American Inc., 2001, pp. 35-43
- [24] J.Heflin and J.Hendler. "A portrait of the Semantic Web in action", *IEEE Intelligent System*, 2001, pp 54-59.
- [25] M. Uschold, "Where are the Semantics in the Semantic Web?" *AI Magazine*, American Association for Artificial Intelligence, pp 25-36
- [26] A. Sheth, C. Ramakrishnan, and C. Thomas, "Semantics for the Semantic Web: The Implicit, the Formal and the Powerful," *Int'l J.Semantic Web and Information Systems*, vol. 1, no. 1, 2005, pp 1-18
- [27] J.Hendler. "Agents and the Semantic Web", *IEEE Intelligent System*, 2001, pp 30-37.
- [28] N. Shadbolt, W. Hall, T. Berners-Lee, "The Semantic Web Revisited", *IEEE Intelligent System*, 2006, pp 96-101.
- [29] E.P. Bontas, L. Nixon, and R. Tolksdorf, "A Conceptual Model for Semantic Web Spaces," *Technical Report B 05-14*, AG Netzbasierte Informationssysteme, Freie Universität Berlin, Germany, 2005.
- [30] Linthicum, D., *Enterprise Application Integration*, Addison-Wesley, New Jersey, 2000.
- [31] Kasunic, M., Anderson, W., "Measuring Systems Interoperability: Challenges and Opportunities" *Technical Note, CMU/SEI-2004-TN-003*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 2004.
- [32] Morris, R., Levine, L., Meyers, C., Place, P., Plakosh, D., "Systems of Systems Interoperability: Final Report" *Technical Report CMU/SEI-2004-TR-004* Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 2004.
- [33] Tolk, A., Muguira, J., "The Levels of Conceptual Interoperability Model", *Proceedings of the 2003 Fall Simulation Interoperability Workshop*, Orlando, FL, 2003
- [34] Ford, T., Colombi, J., Graham, S., Jacques, D., "The Interoperability Score", *Proceedings of the Fifth Annual Conference on Systems Engineering Research*, Hoboken, NJ, 2007
- [35] IEEE, *Standard Glossary of Software Engineering Terminology. Std 610.12-1990*, The Institute of Electrical and Electronics Engineers, New York, 1990.
- [36] O. Corby, R. Dieng-Kuntz and C. Faron-Zucker, "Querying the Semantic Web with the Corese Search Engine," *Proc. 16th European Conf. Artificial Intelligence (ECAI 04)*, IOS Press, 2004, pp. 705-709
- [37] OMG, *Unified Modeling Language: Superstructure*, version 2.1.2, November 2007
- [38] OMG, *OMG Systems Modeling Language (OMG SysML™), V1.0*, available at <http://www.sysml.org>
- [39] Department of Defense Architecture Framework Working Group, *DoD Architecture Framework, version 1.0*. Department of Defense, USA, 2004.
- [40] Lankhorst, M. et al., *Enterprise Architecture at Work*, Springer-Verlag, Berlin, 2005
- [41] D. Chen, G. Doumeingts and F. Vernadat, "Architectures for enterprise integration and interoperability: Past, present and future", *Computers in Industry*, Elsevier B.V., 2008
- [42] E. Johansson and P. Johnson, "Assessment of Enterprise Information Security – Estimating the Credibility of the Results", *Proceedings of the 9<sup>th</sup> IEEE international Annual Enterprise Distributed Object Computing Conference*, The Netherlands, 2005.