Situated Support for Choice of Representation for a Semantic Web Application

Sari E. Hakkarainen, Anders Kofod-Petersen, and Carlos Buil Aranda

Department of Computer and Information Science,
Norwegian University of Science and Technology,
7491 Trondheim, Norway
{sari, anderpe}@idi.ntnu.no, builaran@stud.ntnu.no
http://www.idi.ntnu.no

Abstract. As more and more companies are augmenting their data to include semantics, it is imperative that the choices made when choosing the modelling language are well founded in knowledge about the language and the domain in question. This work extends the Semiotic Quality Framework with computational and situated instruments. Furthermore, it demonstrates how the extended Semiotic Quality Framework can facilitate the choice of the most suited language for a real world application. The application is a directory services system, which currently is being moved into the realms of the Semantic Web.

1 Introduction

The IT industry is currently changing focus from providing storage, processing and network services to provide knowledge intensive information and services to large numbers of customers. The diversity and multitude of resources and applications on the Web places elaborate requirements on methods and tools for efficient generation, manipulation and compositional usage of information and services. Metadata, ontology/domain model and semantic enrichment can bridge the heterogeneity and facilitate the efficient usage of information assets on the Semantic Web [1]. However, a formal, standardised representation of signs and meaning is required [2] for supporting ontologies, i.e. explicit and shared conceptualisations [3] of the domain.

Several general-purpose models for description of Web-resources have emerged, where the intention is to facilitate the search, aggregation, filtering, selection, reasoning and presentation of information assets on, and for the (semantic) Web. However, the number of languages and models is large, as is the number of types of prospective applications. Applications can be categorised according to the kind of domain they address (medical, commerce, education, library, oil drilling, etc), the kind of application they target (knowledge management, process monitoring, archival, etc.) or the kind of modelling environment they are supposed to fit in (taxonomies, data flows, data models, process models, etc.). The span for each of these categories is seemingly endless.

In conceptual modelling there are a number of frameworks suggested for evaluating modelling approaches in general. For instance, the Bunge-Wand-Weber ontology [4] has been used on several occasions as a basis for evaluating modelling techniques,

J. Akoka et al. (Eds.): ER Workshops 2005, LNCS 3770, pp. 386–397, 2005.

e.g. NIAM [5] and UML [6], as well as ontology languages in [7]. The semiotic quality framework first proposed in [8] for the evaluation of conceptual models has later been extended for evaluation of modelling approaches [9] and used for evaluating UML and RUP [10], This framework was also used in evaluation of ontology languages and tools in [11]. Similarly, [7, 12, 13] evaluate various ontology languages. These studies concentrate on evaluating the technical features of representation languages and partly tools, independent of situational factors of particular development projects. Such studies target audience of highly skilled modelling experts rather than the wide spectrum of potential developers of Semantic Web applications. [14] extends and evaluates methods rather than languages. The framework suggested by [15] is meant for requirements specifications, but is still fairly general. There are also more specific quality evaluation frameworks, e.g. [16] for process models, and [17, 13] data/information models.

The objective here is to develop support for the choice of appropriate Web-based knowledge representation formalism. The way-of-working is to 1) evaluate existing representations in general, using an existing semiotic quality framework for conceptual models, 2) to extend the quality framework with computational and situated features, 3) to develop trial ontologies using a common ontology creation tool and the language specifications, and 4) to evaluate the existing representations in an industrial case study. In the case study, the aim is to support the development of an integrated knowledge-based system for directory services by moving from traditional relational data models to semantically richer representations.

The paper is organised as follows: First, an overview of the Semiotic Quality Framework is given. Secondly, the case study is described. This is followed by a discussion of the results obtained. Finally, a conclusion and an outlook on future work is given.

2 The Semiotic Quality Framework

In order to evaluate the Web representation languages, the *Semiotic Quality Framework* (SQF) [9, 11], a model quality framework consisting of five semiotic factors of quality modelling languages, is chosen. The framework has three main characteristics that make it well-suited as an evaluation instrument: 1) it distinguish between goals and means separating what to achieve from how, 2) it is closely related to linguistics and semiotic concepts, and 3) it is based on a constructivist world-view, the framework recognises that models are build from interaction between the designer and the user. The main model of the semiotic quality framework is as follows.

A – **Audience** refer to the individual, A_i, organisational, A_s, and technical actors, A_t who relate to the model. This includes both human participants and artificial actors.

K – **Participant knowledge** is the explicit knowledge that is relevant for the audience A. It is the combined knowledge of all participants in the project.

L – **Language extension** is what can be represented according to the graphical symbols, vocabulary and syntax of the language; the set of all statements that may be informal L_i, semi-formal L_s, or formal L_f.

M – **Model externalization** is the set of all statements in an actor's model of a part of a perceived reality written in a language L.

I – **Social actor interpretation** is the set of all statements which the externalised model consists of, as perceived by the social audience A_i and A_s.

T – **Technical actor interpretation** is all the statements in the conceptual model L as they are interpreted by the technical audience A_t.

D – **Modelling domain** is the set of all statements that can be stated about a particular situation.

The framework evaluates the physical, empirical, syntactic, semantic, pragmatic, perceived semantics, social and knowledge quality; it evaluates the quality of conceptual models, modelling environments, and modelling languages. This work focuses on the evaluation of the Web representations as modelling languages.

2.1 Adapted appropriateness of languages

The Semiotic Quality Framework consists of five quality factors, called appropriateness, namely: Domain Appropriateness (DA), Participant Knowledge Appropriateness (PAK), Knowledge Externalizability Appropriateness (KEA), Comprehensibility Appropriateness (CA), and Technical Actor Interpretation Appropriateness (TAIA). Here we modify the DA as in [18], as follows.

DA covers seven perspectives for languages: 1) Structural Perspective refers to the static structure, classes and properties, 2) Functional Perspective refers to the processes, activities, and transformations, 3) Behavioural Perspective refers to the states and transitions between them, 4) Rule Perspective refers to the rules for certain processes, activities, and entities, 5) Object Perspective refers to the resources, processes and classes, 6) Communication Perspective refers to the language actions, meaning and agreements, and 7) Actor and Role Perspective refers to the actor, role, society and organisation.

With the modification of the DA we acquire the elements needed to analyse the most practical features of the languages. With the PKA we measure the knowledge of the user. With the KEA we analyse if the language provides enough elements to represent the domain model specified. With CA we analyse if the language is consistent enough and provides clear elements for modelling the domain, and with TAIA we analyse if the language provides enough features for allowing automatic reasoning, the key concept in our investigation. The quality factors are further developed in the sequel.

2.2 Selection Criteria for Quality Factors

For the quality of conceptual modelling languages, Sindre [19] identifies criteria for the constructs of the language and how these constructs are presented visually. Four main groups of sub-criteria are identified: perceptibility, expressive power, expressive economy, method tools and potential. Seltveit [20] adds the criteria of reducibility, meaning the features provided by the model to handle large and complex models.

Let $\mathcal{CF}$ be an evaluation framework such that $\mathcal{CF}$ has a fixed set $\mathcal{A}$ of appropriateness categories a, where $\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5\}$ and $a_i \in \mathcal{A}$. Each a is a quadruple $< id, \quad descriptor, \quad C, \quad cw >$, where id is the name of the category, $descriptor$ is a natural language description, C is a set of selection criteria ac, and cw defines a function of S that return -1, 1 or 2 as coverage weight, where S is a set of satisfied elements ac in the selection criteria C of each appropriateness category in $\mathcal{A}$. Intuitively, we define a number of selection criteria alongside an associated coverage weight function for each category in the evaluation framework. The appropriateness categories with attached descriptors, selection criteria and coverage weight functions are as follows.

a_1 – *Domain appropriateness (DA)* indicates whether the method guidelines address the problems of eliciting/representing relevant facts of the problem domain. Ideally, $D \backslash L = \emptyset$, i.e. there are no statements in the expected application domain that cannot be expressed in the target language, and one should not be guided to express things that are not in the domain (limited number of constructs). The former criterion means that a_1c_1 - the developer is guided to make use of high expressive power whereas the latter means that a_1c_2 - there is a limited number of modelling constructs that are generic, composable and flexible in precision. The equation 1 holds for each modelling perspective of a_1p_1 - structural (SP), a_1p_2 - functional (FP), a_1p_3 - behavioural (BP), a_1p_4 - object (OP), a_1p_5 - communication (CP), and a_1p_6 - actor-role (AP) perspective.

a_2 – *Participant knowledge appropriateness (PKA)* indicates whether the method corresponds to what participant in the modelling activity perceive as a natural way of working. Ideally, $K \cap L \backslash L = \emptyset$, that all the statements in the models of the languages used by the participants are part of their explicit knowledge. Hence a method guideline a_2c_1 - should not promote usage of statements not in a participant's knowledge, a_2c_2 - external representation should be intuitive, and a_2c_3 - non-intuitive representations should be introduced carefully.

$$cw_1(S_1) = \begin{cases} 2 & if \quad a_1c_1 \wedge a_1c_2 \in S_1 \\ 1 & if \quad a_1c_1 \vee a_1c_2 \in S_1 \\ -1 & if \quad S_1 = \emptyset \end{cases} \qquad (1)$$

$$cw_2(S_2) = \begin{cases} -1 & if \quad |S_2| = 0 \\ 1 & if \quad 0 < |S_2| \leq 1 \\ 2 & if \quad 2 < |S_2| \leq 3 \end{cases} \qquad (2)$$

a_3 – *Knowledge externalization appropriateness (KEA)* indicates whether the method assists the participants in externalising their knowledge. $K \cap L \backslash K = \emptyset$, i.e. there are no statements in the explicit knowledge of the participant in the modelling activity that cannot be expressed in the target language. This appropriateness focuses on how relevant knowledge may be articulated in the language rather than what knowledge is expressed. This implies the partial quality goals of generality, a_3c_1 – the guidance to use the language should be as domain independent as possible, and completeness a_3c_2 – there is guidance for all possible usages of the language.

a_4 – *Comprehensibility appropriateness (CA)* indicates whether the participants are able to comprehend the method guidelines. Ideally, $L \backslash I = \emptyset$, i.e. all the possible statements of the language are understood by the participants in the modelling effort using the method guidelines. Thus, a_4c_1 - the described modelling constructs are easily distinguished from each other, a_4c_2 - the number of constructs is reasonable or organised in a natural hierarchy, a_4c_3 - proposed use of modelling constructs is uniform for all the statements expressed in the target language, a_4c_4 - the guidance is flexible in the level of detail in the target language, and a_4c_5 - separation of concerns and multiple views is supported.

$$cw_3(S_3) = \begin{cases} 2 & if \quad a_3c_1 \wedge a_1c_2 \in S_3 \\ 1 & if \quad a_3c_1 \vee a_1c_2 \in S_3 \\ -1 & if \quad S_3 = \emptyset \end{cases} \quad (3) \qquad cw_4(S_4) = \begin{cases} -1 & if \quad 0 < |S_4| \leq 1 \\ 1 & if \quad 1 < |S_4| \leq 3 \\ 2 & if \quad 3 < |S_4| \leq 5 \end{cases} \quad (4)$$

a_5 – *Technical actor interpretation appropriateness (TAIA)* indicates whether the method guidelines lend themselves to automated tool support or assist in support for reasoning. Ideally, $T \backslash L = \emptyset$, all possible mechanisms in the technical participants interpretation are supported by the target language. This implies the partial quality goals for automatic reasoning support in the instructions provided for the target language, i.e. a_5c_1 - both formal syntax and semantics are operational and/or logical, a_5c_2 - efficient reasoning support is provided by executability, a_5c_3 - natural language reasoning is supported, and a_5c_4 - information hiding constructs are provided enabling encapsulation and independent components.

$$cw_5(S_5) = \begin{cases} 2 & if \quad a_5c_1 \wedge (a_5c_2 \vee a_5c_3 \vee a_5c_4) \in S_5 \\ 1 & if \quad a_5c_1 \vee a_5c_2 \vee a_5c_3 \vee a_5c_4 \in S_5 \\ -1 & if \quad S_5 = \emptyset \end{cases} \quad (5)$$

The selection criteria for the appropriateness categories above are exhaustive in the categories a_2, and a_4, whereas the set of satisfied criteria S of the remaining categories may also be the empty list. None of the criteria are mutually exclusive. The coverage weight cw is independent of any category-wise prioritisation. Since the intervals are decisive for the coverage weight they can be adjusted depending on preferences of the evaluator. However, when analysing different evaluation occurrences the intervals need to be fixed in comparison, but may be used as dependent variable.

2.3 Weighted Quality Requirements

Here, we adopt the PORE methodology [21] to prioritise the classification criteria based on company's requirements in order to evaluate the ontology building guidelines in this particular situation. The method has been applied successfully on SQF in [14, 22] for method guideline evaluation and classification, respectively. Hence, the importance weights for each appropriateness category are calculated as follows.

Let $R(CF)$ be a set of weighted requirements such that R has a fixed set $R\mathcal{A}$ of categories ra, where categories in $R\mathcal{A}$ correspond with categories $\mathcal{A}$ of an evaluation framework EF, i.e. $R\mathcal{A} = \mathcal{A}$, and $a \in \mathcal{A}$, $ra \in R\mathcal{A}$. ra is a triple $< id, \quad descriptor, \quad iw >$, where id is the name of the appropriateness requirement category, $descriptor$ is a natural language description of the appropriateness requirement, and iw_{ra} defines a function of I that returns 0, 3, or 5 as importance weight based on priorities and policy of the company, where I is a set of importance judged elements ra in the selection criteria C of each category in $R\mathcal{A}$.

$$iw_{ra}(I) = \begin{cases} 1 & if \quad ra \quad is \quad optional \\ 3 & if \quad ra \quad is \quad recommended \\ 5 & if \quad ra \quad is \quad essential \end{cases} \quad (6)$$

Table 1. Requirements

Appropriateness	Importance	*iw*
Domain Appropriateness	High	5
Participant Knowledge Appropriateness	Medium	3
Knowledge Externalizability Appropriateness	Low	1
Comprehensibility Appropriatenss	Low	1
Technical Actor Interpretation Appropriateness	high	5

3 An Industrial Semantic Web Application

Our industry partner proposed an investigation of directory services. The problem consist of two databases containing the directory service data of Sweden and Norway. This means millions of records with information about people, streets, companies, and the respective country. Further, it is difficult to obtain accurate reasoning mechanisms or results from queries based on the databases solely.

The company wants to move into the emerging Semantic Web technologies. The investigation consists of creating ontologies that represent Norwegian and Swedish databases based on the databases schemata. The ontologies are created in three different ontology languages. RDF(S) [23] for analysing if the expressiveness of this language is enough for our case and OWL [24] and Topic Maps [25], two more complex ontology languages which offer more facilities for representing data in a proper way. The SQF representation requirements of the company are summarised in Table 1.

The databases were in different formats. The one containing the Swedish data was schematised in XML and the DTD was given, making it easier to approach the creation of the ontologies. The other, containing the Norwegian data, was schematised in a text document without a formal schema. Here, a reverse engineering process was required.

In order to obtain more objective basis for the analysis, the ontologies were 1) created using the [26] ontology building method, 2) given scope according to a common UML representation as a control ontology, and 3) implemented using Protégé 2000 as the editing tool.

3.1 UML Models Based on the Databases

A consolidated UML model for the two national databases is depicted in Figure 1. The UML model of the Swedish database is extracted from the DTD and the interpretations of each class are tailored while creating the model. The classes are used for representing the information of persons in the database. There is no distinction between persons and companies in the Swedish data.

The Norwegian data was provided in a text format, without a structured specification of classes, relations or attributes. As a re-engineering effort, we extracted a basic model from the file.

The Norwegian database model is an approximation to the original database. The original data was not considered definitive and was not adequate to build an ontology from. Thus, some attributes have been added and the structure of the database has been re-designed in order to be similar to the Swedish database. Finally, the consolidated

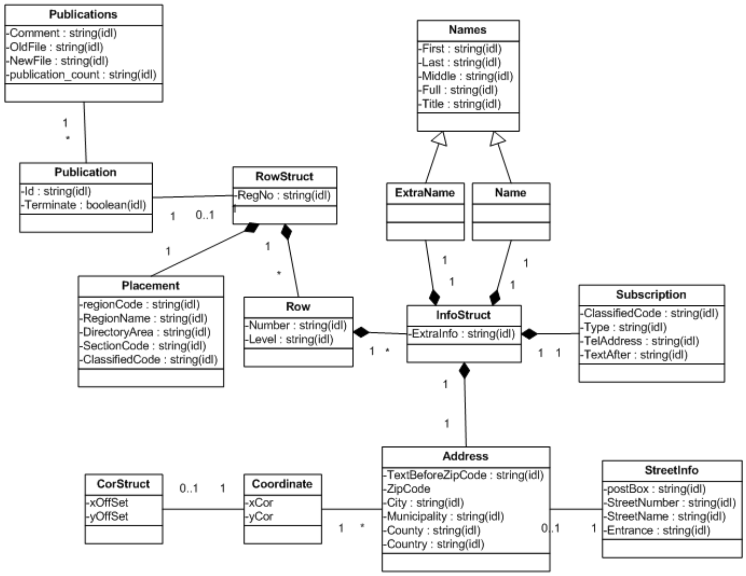

Fig. 1. UML model of the Swedish data

UML model consists of 14 classes, 14 relations, and 38 attributes. This model is used as control model for creating ontologies in the below representations.

RDF(S) ontology. Almost all the classes in the RDF(S) ontology are the same as in the UML model in Figure 1. The classes are: Publications, Publication, Row, Address, Subscription, Placement, StreetInfo, Coordinate and the two kinds of names with the superclass Names. The only classes missing are RowStruct and InfoStruct. These two classes are not necessary in RDF(S) because we are able to specify the domain and the ”Allowed classes” for the instances properties. All the properties assigned to this slot will be applied to all its elements. The disadvantage of this method is that we have to create a class for the attribute RegNo since we want to apply the property to the attribute.

OWL ontology. The UML model represented in Figure 1 can be fully designed within OWL. All the classes and properties in the OWL ontology have a corresponding class or relationship in the OML model. OWL offers enough facilities to represent this model easily. Furthermore, we can, unlike within RDF(S), add more relations and restrictions to these relations. The model created is the same as the model created with RDF(S) and the model specified in UML.

Topic Maps ontology. The hierarchy of classes created in Topic Maps is almost the same as the one created within OWL and RDF(S). In the Topic Maps ontology there are the Name superclass with the children Name and ExtraName, Publications, Placement, Row and the other relevant classes extracted from the UML model.

The main difference is the parent class of every class. This parent is the Topic superclass. All classes of our model inherit the properties of Topic, and we have a better organisation of our model. The attributes (slots or facets), which can be just a typical slot typed as Integer or String or an instance of any class, can be defined in two different ways. Just like a simple slot or like a Topic Name, this allows us to create more specific relations. To create associations we follow the same way as the one used to create the classes or the attributes. First creating an instance of a Topic, and then referencing the association to the Topic or creating attributes like instances, as with OWL or RDF(S).

4 Comparative Evaluation of Representations

The adapted SQF evaluation model provides an instrument to evaluate quality of Web representation languages. Below, we summarise the evaluations of RDF(S), OWL and Topic Maps in the general terms, independent of target application. In the evaluation table (Table 2), the columns are the various appropriateness levels from section 2.2, NL description of the coverage, and the assigned criteria weights.

RDF(S). The table shows that RDF(S) provides the basic elements in order to satisfy the Domain Appropriateness requirements, because it covers the Structural Perspective fully and partially the Object Perspective and gets a weight of 1. The other appropriateness are covered by RDF(S) only providing the basic requirements of these appropriateness and therefore have the weight of 1.

OWL. The table shows that OWL provides the basic elements in order to satisfy the Domain Appropriateness requirements, because it covers the Structural Perspective and Rule Perspective fully and partially the Object Perspective and gets a weight of 2. The Technical Actor Appropriateness is fully covered because OWL is designed to support this appropriateness and gets a weight of 2. The other appropriateness are covered by OWL only providing the basic requirements of these appropriateness, thus they receive a weight of 1.

Topic Maps. The table shows that Topic Maps provides the basic elements to satisfy the Domain Appropriateness requirements, because it covers the Structural Perspective fully and partially the Object Perspective and gets a weight of 1. Topic Maps also covers Technical Actor Appropriateness because it provides more elements to detailed data and facilitate the automatic reasoning by the information agents and gets a weight of 2. The other appropriateness are covered by Topic Maps only providing the basic requirements of these appropriateness and receives a weight of 1.

4.1 Comparison of the Three Languages

After this analysis of the languages in Tabel 2, it is possible to compare with the quality requirements of the cooperating company. First, the total coverage weights Tw_i for each representations i are calculated. In Table 3, we have summarised which of the studied ontology building methods that meet the situated, quality-based requirements specified by our industry partner. Here, the importance weights from Table 1 are multiplied by the coverage weights from Table 2 and total weights calculated using Equation 7 are used as overall feasibility rate for supporting the choice of ontology building guidelines.

$$Tw_i = \sum_{ra \in \mathcal{A}} (cw_{ra} \times iw_{ra}) \tag{7}$$

The weights assigned in Table 1 are based on the requirements specified in Section 3 , in Table 1. The requirements are specified in natural language. We have translated these requirements to a more proper representation in order to make a comparison with results obtained from the language analysis. The situated comparison of the languages with the requirements are shown in the Table 3.

In Table 3 we have added a new column where it is specified the final weight of the languages. We have multiplied the results obtained for each language by the weight of the appropriateness given in the Table 1. With this new column and the new weights

Table 2. Analysis of the three languages

Appropriateness	Description	*cw*
	RDF(S)	
DA	RDF(S) covers structural perspective. Objective perspective is partially covered. Does not cover rule, behavioural, functional, actor or role perspective	1
PKA	Dependent on the desinger's experience	-1
KEA	Posibility to model determinated situations	1
CA	RDF(S) elements are easily distinguished. The number of phenomena are reasonable. The structure of RDF(S) is partially consistent.	1
TAIA	RDF(S) partially covers this appropriateness. Provides basic elements for automatic reasoning.	1
	OWL	
DA	OWL covers structural and rule perspective. Objective perspective is partially covered. Does not cover behavioural, functional, communication, actor, or role perspective.	2
PKA	Dependent on the desinger's experience	-1
KEA	Domain dependent. Possibility for modelling main database concepts.	-1
CA	OWL elements are easily distinguished. The number of phenomena is reasonable. Symbol discrimination is not fully covered. The structure of OWL is consistent.	1
TAIA	OWL covers this appropriateness.	2
	Topic Maps	
DA	Topic Maps covers structural perspective and partially object perspective. Does not cover functional, behavioural, rule, object, communication, actor or role perspective.	1
PKA	Dependent on the desinger's experience	-1
KEA	Domain dependent	-1
CA	Topic maps diferentiates between symbols. The number of phenomena are reasonable, but less reasonable than RDF(S) or OWL. The structure is partially consistent. Not enough expressive economy.	1
TAIA	Topic Maps covers this appropiateness.	2

Table 3. Situated comparison of languages

Appropriateness	RDF(S)	OWL	Topic Maps
DA	$1x5$	$2x5$	$1x5$
PKA	$-1x3$	$-1x3$	$-1x3$
KEA	$-1x1$	$-1x1$	$-1x1$
CA	$1x1$	$1x1$	$-1x1$
TAIA	$1x5$	$2x5$	$2x5$
Total weight	7	17	10

added we can distinguish more easily the most adequate language and eventually elaborate with slightly modified importance weights.

Thus, the most appropriate language in our case, is the Web Ontology Language, OWL. This is due to the elements that it provides for creating first order predicates. The other languages do not offer this feature because RDF(S) is not designed to offer these facilities and Topic Maps is orientated to create a better quality relationships between its elements but does not provide enough elements for constraining these relations.

5 Conclusions and Future Work

An evaluation of representations for Semantic Web applications was conducted adapting the extended [14, 22] Semiotic Quality Framework (SQF) [9, 11]. The comparative analysis was performed in two steps, one general evaluation, i.e. their applicability for Semantic Web applications in general, and one contextual, i.e. how appropriate they are for ontology development in a real world project in particular. The applicability of situated SQF was tested in a case study. The main results are as follows.

- SQF is suited for evaluating semantic web representations. Use of the numerical values for the weights and adoption of the PORE methodology [21] produce explicit evaluation results.
- In both steps, the general classification and the evaluation against the situated requirements, OWL [24] came out on top, meeting many of the evaluation criteria. This is also the only representation which support the rule perspective.
- Following OWL, Topic Maps [25] proved to be slightly better than RDF(S) [23].

The contribution of this paper is two-fold: First, an existing evaluation framework was tried out with other evaluation-objects than it has been used for previously; Second, numerical values and metrics were incorporated to the quality factors of SQF and thus, supporting qualification of weighted selection. The case study suggests that, given the small adjustments, the framework originally intended for evaluation of conceptual modelling languages, is applicable in evaluation of semantic Web representations, regardless if the analysis is used for their selection, quality assurance, or engineering

The concrete ranking of methods may be of limited use, as new ontology languages and methodologies and associated tools are developed, and as the existing languages evolve. Nevertheless, it can be useful in terms of guiding the current and future creators of such languages and modelling environments. Drawing attention to the weakness of current proposals, these can be mended in future proposals, so that there will be several high quality representations to choose from in the future. The underlying assumption for our work is that the option to choose appropriate representation suited for any application at hand may increase and widen the range and scalability of the Semantic Web ontologies and applications.

There are several interesting topics for future work, such as supplementing the theoretical evaluations with empirical ones as larger scale semantic Web applications arise utilising the empirical nature of [9], as well as evaluating more representations as they emerge, e.g. OWL-S and SWRL. Further possibilities are in investigating the appropriateness of the formalisation quality criteria in the Unified methodology [2] as a complement to the semiotic quality framework in order to conduct evaluation of the process oriented methodological frameworks that were out of scope of this study.

Acknowledgement

We would like to thank the company Invenio for their cooperation and invaluable support in conducting this research.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American (2001)
2. Uschold, M., Grüninger, M.: Ontologies: Priciples, methods, and applications. Knowledge Engineering Review **11** (1996) 93–155
3. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition **5** (1993) 199–220
4. Wand, Y., Weber, R.: Mario Bunge's Ontology as a formal foundation for information systems concepts. In Weingartner, P., Dorn, G., eds.: Studies on Mario Bunge's Treatise. Rodopi (1990)
5. Weber, R., Zhang, Y.: An analytical evaluation of NIAM's grammar for conceptual schema diagrams. Information Systems Journal **6** (1996) 147–170
6. Opdahl, A.L., Henderson-Sellers, B.: Ontological evaluation of the UML using the Bunge-Wand-Weber model. Software and Systems Modelling (SoSyM) **1** (2002) 43–67
7. Davies, I., Green, P., Milton, S., Rosemann, M.: Using Meta-Models for the Comparison of Ontologies. In: Proceedings of the Eighth CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD'03). (2003)
8. Lindland, O.I., Sindre, G., Sølvberg, A.: Understanding Quality in Conceptual Modeling. IEEE Software **11** (1994) 42–49
9. Krogstie, J., Sølvberg, A.: Information systems engineering - Conceptual modeling in a quality perspective. Kompendiumforlaget, Trondheim, Norway (2003)
10. Krogstie, J.: Using a Semiotic Framework to Evaluate UML for the Development of Models of High Quality. In Siau, K., Halpin, T., eds.: Unified Modeling Language: System analysis, design, and development issues. IDEA Group Publishing (2001)
11. Su, X., Ilebrekke, L.: A comparative study of ontology languages and tools. In Pidduck, A.B., Mylopoulos, J., Woo, C.C., Ozsu, M.T., eds.: Advanced Information System Engineering, 14th International Conference, CAiSE 2002, Toronto, Canada, May 27-31, 2002, Proceedings. Volume 2348 of Lecture Notes in Computer Science., Springer Verlag (2002) 761–765

12. Gómez-Péres, A., Corcho, O.: Ontology Languages for the Semantic Web. IEEE Intelligent Systems (2002) 54–60
13. Schuette, R.: Architectures for evaluating the quality of information models – a meta and an object level comparison. In Akoka, J., Bouzeghoub, M., Comyn-Wattiau, I., Métais, E., eds.: Proceedings of 18th International Conference on Conceptual Modeling, Paris, France. Volume 1728 of Lecture Notes in Computer Science., Springer Verlag (1999)
14. Hakkarainen, S., Hella, L., Strasunskas, D., Tuxen, S.: Choosing Appropriate Metod Guidelines for Web-Ontology Building. In: Proceedings of the 24th Internation Conference on Cenceptual Modeling (ER 2005). (2005) To appear.
15. Pohl, K.: Three dimensions of requirements engineering: a framework and its applications. Information Systems **19** (1994) 243–258
16. Becker, J., Rosemann, M., von Uthmann, C.: Guidelines of Business Process Modeling. In Aalst, W., Desel, J., Oberweis, A., eds.: Business Process Management, Models, Techniques, and Empirical Studies. Volume 1806 of Lecture Notes In Computer Science., Springer Verlag (2000) 30–49
17. Moody, D.L., Shanks, G.G., Drake, P.: Evaluating and Improving the Quality of Entity Relationship Models: Experiences in Research and Practice. In Ling, T.W., Ram, S., Lee, M.L., eds.: Proceedings of the 17th International Conference on Conceptual Modeling, Singapore (ER '99). Volume 1507 of Lecture Notes in Computer Science., Springer Verlag (1998)
18. Su, X., Ilebrekke, L.: 14. In: Using a Semitic Framework for a Comparative Study of Ontology Languages and Tools. IDEA Group Publishing Group (2004) 278–299
19. Sindre, G.: Hicons: a general diagrammatic framework for hierarchical modelling. PhD thesis, Norwegian Institure of Technology (1990)
20. Seltveit, A.H.: Complexity reduction in information systems modelling. PhD thesis, Norwegian Institute of Technology (1994)
21. Maiden, N.A., Ncube, C.: Acquiring cots software selection requiremtents. IEEE Software **15** (1998) 46–56
22. Hakkarainen, S.E., Strasunskas, D., Hella, L., Tuxen, S.M.: Weighted evaluation of ontology building methods. In: CAiSE 2005 Forum. (2005) To Appear.
23. Brickley, D., Guha, R.V., eds.: RDF Vocabulary Description Language 1.0: RDF Schema. W3C (2004)
24. McGuinness, D.L., van Harmelen, F., eds.: OWL Web Ontology Language Overview. W3C (2004)
25. Biezunski, M., Bryan, M., Newcomb, S.R., eds.: ISO/IEC 13250:2000 Topic Maps: Information Technology – Document Description and Markup Language. International Organization for Standardization (1999)
26. Noy, N.F., McGuinness, D.L.: Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Stanford Knowledge Systems Laboratory (2001)