Chapter 7

Concluding Remarks

Realizing the benefits that Semantic Web technologies offer, many traditional WIS design methodologies as well as newly proposed WIS design methodologies use Semantic Web technologies for modeling WIS. Hera is a new design methodology that targets SWIS design. It distinguishes two phases: the data collection phase and the presentation generation phase. The first phase makes available data coming from different, possibly heterogeneous data sources. The second phase produces Web presentations tailored to the user and its browsing platform. In this dissertation we have presented the presentation generation phase of Hera. Section 7.1 sums up our results. Section 7.2 suggests future research directions based on our results.

7.1 Conclusions

At the beginning of the dissertation, in Chapter 1, we did ask five research questions. In the rest of this section we will summarize our findings and give the answers to the five research questions that we came up with in the different chapters of this dissertation.

Question 1: How to design the presentation generation for SWIS?
In Chapter 2 we studied several (S)WIS design methodologies. Among the found characteristics of these methodologies we mention: data integration, user interaction, presentation modeling, presentation personalization etc. At the current moment SWIS design methodologies are at their infancy, as we found no SWIS design methodology that has (all) the previously identified characteristics. In order to fill this gap, in Chapter 3 we proposed Hera, a SWIS methodology that has many of the features identified in Chapter 2. This dissertation concentrates on the presentation generation phase of the Hera methodology.

The presentation generation phase of the Hera methodology identifies the following design steps:

- **conceptual design**: constructs the conceptual model (CM), a uniform representation of the application's data. It defines the concepts and the concept relationships that are specific to the application's domain.

- **application design**: constructs the application model (AM), the navigational structure over the application's data. It defines slices and slice relationships. A slice is a meaningful presentation unit of some data. There are two types of slice relationships: slice navigation, used for links between slices, and slice aggregation, used for embedding a slice into another slice.

- **presentation design**: constructs the presentation model (PM), the look-and-feel specifications of the presentation. It defines regions and region relationships. As for slices, there are two types of region relationships: region navigation, used for links between regions, and region aggregation, used for embedding a region into another region. Regions have associated layout (positioning of inner regions inside a region) and style (fonts, colors etc.) information.

For the model representations we used RDF, the foundation language of the Semantic Web. Some of the advantages of using RDF as a representation language are: it is able to describe semi-structured Web data, it enables the reuse of previously defined vocabularies (e.g., the CC/PP UAProf vocabulary to represent user preferences and device capabilities), it allows the exchange of data between applications in a uniform format etc.

Question 2: How can we support adaptation during the design of the presentation generation for SWIS?

In Chapter 3 we have identified two types of adaptation that can be supported in the presentation generation phase of Hera:

- **static adaptation**: adaptation performed before the user starts browsing the presentation. The static adaptation is specified by appearance conditions, for elements in the CM, AM, and PM. These conditions use data from a user profile (UP) which stores the static user preferences and device capabilities. Elements for CM, AM, and PM that have the associated conditions not satisfied are removed from the specifications.

- **dynamic adaptation**: adaptation performed during the user browsing before each page is generated. The dynamic adaptation uses AM queries in order to update the User Session (US). US stores dynamic data, i.e., data created during user browsing based on user's input. The presentation generation phase uses data from US in a similar way as the CM.

Based on these two types of adaptation, in Chapter 3, we have presented two variants of Hera's presentation generation phase: the static variant and the dynamic variant. In the specifications of the dynamic variant we use (Se)RQL one of the most expressive RDF query languages. At the current moment W3C started to work at the RDF query language called SPARQL [Prud'hommeaux and Seaborne, 2005]. When this language will become more mature we plan to use it also in Hera as a replacement for (Se)RQL.

Question 3: What CASE tools can support the design of the presentation generation for SWIS?
One of the characteristics of SWIS design methodologies identified in Chapter 2 was the tool support. SWIS design methodologies are not well supported by CASE tools. In order to better sustain the design activities proposed in the presentation generation phase of the Hera methodology we have implemented a CASE tool, the Hera Presentation Generator (HPG), which is described in Chapter 4. It integrates several tools built in the last couple of years in the Hera project: builders for CM, AM, and PM, a prototyping tool based on previously defined models, and a data transformations visualization tool.

There are two variants defined for HPG: HPG-XSLT, which corresponds to the static variant of the Hera presentation generation phase, and HPG-Java, which corresponds to the dynamic variant of the Hera presentation generation phase. HPG-XSLT implements the data transformations using XSLT stylesheets, and HPG-Java implements the data transformations in Java using Jena and Sesame libraries. HPG-Java exploits more of the RDF model semantics than HPG-XSLT. Nevertheless, HPG-Java lost the declarativity, simplicity, and reuse capabilities of the XSLT transformation templates. A distributed architecture of the HPG based on Web Services is also provided.

Question 4: How can one realize query optimization inside a SWIS?
The dynamic variant of Hera's presentation generation phase uses RDF queries. The execution time of these RDF queries by a query engine is an important factor in the SWIS response time to a user request. In Chapter 5 we have proposed RAL, an RDF algebra that can be used for RDF query optimization. It defines a data model and a set of operators. The collections (sets) of nodes are closed under all operators. There are two types of operators: extraction operators, which retrieve nodes of interest from an input collection, and construction operators that build an output model possibly using also the extracted nodes. Some of the extraction operators were inspired by the ones found in relational algebra.

RAL operators satisfy equivalence laws, some of them resembling the relational algebra equivalence laws. We propose a heuristic algorithm for RDF query optimization similar to the one given in the relational algebra (i.e., pushing the selections/projections down as far as possible, and applying the most restrictive selections first). A translator from an RDF query language like (Se)RQL to RAL and a RAL engine that implements this query optimization algorithm can be a replacement of the currently used RDF query engines that do not support query optimization.

Question 5: What are suitable visualization techniques for the data used by a SWIS?
All Hera models and their instances are described in RDF. Model instances and to some extent even models do form large graphs for which visualization techniques can be useful to get a better insight into the model properties. In Chapter 6 it is described how one can apply a general-purpose graph visualization engine, GViz, for the visualization of the models used in the presentation generation phase of Hera. We did visualize conceptual models, conceptual model instances, application models, and application model instances.

One of the main advantages of GViz compared with other graph visualization tools is its customization facilities. We did define for example specific glyphs and layouts, for the visualization of the Hera models. Based on the script-based customization mechanism we were able to produce in a matter of minutes model-specific visualization scenarios.

7.2 Future Research

This dissertation describes the presentation generation phase of a SWIS design methodology. There are several directions in which this work can be extended.

One research direction is to extend the presented methodology with new steps and models. The following extensions are possible:

- To extend the proposed SWIS design methodology with a requirements gathering phase. Previous work done for WIS design methodologies based on use case specifications, user interaction specifications (OOHDM [Guell et al., 2000]), and task modeling (WSDM [De Troyer and Casteleyn, 2004]) can be useful for this purpose. By devising an interaction digram or a task model one can automatically generate the navigation structure of the application eliminating the design effort of building AMs. The generated AM can be further customized by the designer.

- To extend the dynamic adaptation in such a way that a SWIS produces adaptive hypermedia. We have published some ideas in this direction in [Vdovjak et al., 2003]. Based on these ideas one could dynamically build the User Model, Domain model, and Adaptation Model of AHAM. In this way one could also reuse existing adaptive hypermedia engines (like AHA! [De Bra et al., 2000]) inside an adaptive SWIS.

- To define a declarative RDF transformation language to be used for the data transformation specifications in the proposed methodology. We envisage that this language will be based on templates similar to XSLT but applied to the RDF context.

- To extend the Web service-oriented architecture of the HPG with new services like a data query service, a data integration service, or a service able to generate adaptive hypermedia presentations.

- To use richer Semantic Web languages for model specifications. The RDF extensions that we added for the conceptual model vocabulary (like the inverse and cardinality of relationships) are already part of the OWL language. In this way the applications built with the Hera methodology would have an even higher degree of interoperability with other applications.

Another research direction is related to RAL. The following research activities are possible:

- To build a translator from (Se)RQL or SPARQL to RAL and a RAL engine that implements our query optimization heuristics. The experiences that we could gain by using these two tools can help us in refining RAL so that it better meets practical needs.

- To explore new equivalence laws possibly involving also the construction operators. In addition, one could also use for query optimization the semantic equivalence laws that are valid in specific RDF models.

Another possible direction is related to the RDF graph visualization techniques. Some possible research activities are:

- To explore 3D visualization of RDF data, possibly getting an even better insight into its structure. The previous experiences with using GViz [Telea et al., 2002] for the 3D visualization of the graphs involved with software (re)engineering might be also useful in the RDF context.

- To use RDF graph visualization in conjunction with an RDF query language (like SPARQL) for depicting graphically the input and output sets of an RDF query. This will help for example the user to visually identify which input nodes and edges were used in the output of a query.

- To conduct visualization experiments with other Semantic Web languages like OWL. As indicated previously we plan to user richer (than RDF) Semantic Web languages for model specifications. As such it will be interesting to visualize the Hera models represented in these languages.

Bibliography

De Bra, P., Aerts, A., Houben, G. J., and Wu, H. (2000). Making general-purpose adaptive hypermedia work. In *WebNet 2000 World Conference on the WWW and Internet (WebNet 2000)*, pages 117–123. AACE.

De Troyer, O. and Casteleyn, S. (2004). Designing localized web sites. In *5th International Conference on Web Information Systems Engineering (WISE 2004)*, volume 3306, pages 547–558. Springer.

Guell, N., Schwabe, D., and Vilain, P. (2000). Modeling interactions and navigation in web applications. In *Conceptual Modeling for E-Business and the Web (ER 2000) Workshops*, volume 1921 of *Lecture Notes in Computer Science*, pages 115–127. Springer.

Prud'hommeaux, E. and Seaborne, A. (2005). Sparql query language for rdf. W3C Working Draft 17 February 2005.

Telea, A., Maccari, A., and Riva, C. (2002). An open toolkit for prototyping reverse engineering visualization. In *IEEE EG VisSym '02*, pages 241–250. Eurographics.

Vdovjak, R., Frasincar, F., Houben, G. J., and Barna, P. (2003). Engineering semantic web information systems in hera. *Journal of Web Engineering*, 2(1-2):3–26.