Chapter 1

Introduction

Since its birth in the early nineties the World Wide Web has grown into one of the most popular channels for reaching a very diverse audience on different platforms worldwide and 24 hours per day. Its success is overwhelming and its impact on the humankind is tremendous. Some compare its importance with Gutenberg's invention of the printing press. As a result of Web popularity, many information system have been made available through the Web, resulting in so-called *Web Information Systems* (WIS)[1] [Isakowitz et al., 1998].

The early WIS presented information in terms of carefully authored hypermedia documents. This approach fails to meet the growing demand to make available on the Web large amounts of data. The *data-intensive* aspect of WIS is present in many applications which can be found today on the Web: institutional portals, community Web sites, online shops, digital libraries, etc. Going from data to a user appealing Web presentation is a complex process that asks for a highly structured and controlled approach to WIS design. Conventional *software engineering* practices are useful for the design of the back-end of these applications. The hypermedia paradigm specific to the Web application front-end asks for a Web-specific engineering approach. A new emerging discipline called *Web engineering* [Murugesan et al., 2001] is concerned with the establishment of systematic approaches to the development of WIS.

A typical scenario in a WIS is the following: in response to a user query the system (semi-)automatically generates a hypermedia presentation. The content of the hypermedia presentation is gathered from different, possibly heterogeneous, sources that are distributed over the Web. A characteristic aspect of a WIS is the *personalization* of the generated hypermedia presentation. The one-size-fits-all approach that is so typical for traditional hypermedia is not suitable for delivering information at run-time to different users with different platforms (e.g., PC, PDA, WAP phone, WebTV) and different network connections (e.g., dial-up modem, network copper cable, network fiber optic cable). The WIS support for user interaction (e.g., by means of forms) enables the user to influence the generated hypermedia presentation based on his explicit needs.

[1] From now on we will refer by WIS to both Web Information Systems and a Web Information System, as a matter of convenience.

Several methodologies have been proposed for the design of WIS. In the plethora of proposed methodologies we distinguish the *model-driven methodologies* (e.g., WebML, OOHDM, RMM, UWE, OO-H, OOWS, OntoWebber) that use models to specify the different aspects involved in the WIS design. The advantages of such model-based approaches are countless: better understanding of the system by the different stakeholders, support for reuse of previously defined models, checking validity/consistency between different design artifacts, (semi-)automatic model-driven generation of the presentation, better maintainability, etc.

The next generation Web, the *Semantic Web* [Berners-Lee et al., 2001], is an extension of the current Web in which data will have associated semantics to it. Several Semantic Web languages (e. g. RDF(S), OWL) have been proposed to represent data semantics (as metadata) in a uniform way at different abstractions levels. WIS that use Semantic Web technologies we call Semantic Web Information Systems (SWIS). The Semantic Web will enable the interoperability between different SWIS. The benefits that the Semantic Web brings to SWIS are remarkable: a large amount of annotated data accessible by any SWIS, exchange and reuse of data models between different SWIS, flexible representation of the Web semistructured (meta-)data, etc. Model-driven SWIS design methodologies that exploit the advantages of the Semantic Web, will help the construction of successful SWIS on the future Web.

1.1 Motivation

Despite the existence of many model-driven methodologies for designing WIS there are not many model-driven methodologies targeting the design of SWIS. The existing SWIS design methodologies (e. g. XWMF, SEAL, OntoWebber, SHDM) fail to support the complete design process of a SWIS. Personalization, a critical aspect in a SWIS, is very often neglected in present SWIS design methodologies. Also, the ability to model the user interaction with the system (besides the 'link following' mechanism) is missing from these methodologies.

A methodology that comes with CASE tools will be better accepted by WIS developers. Most of the existing CASE tools (e.g., WebRatio, OOHDM-Web, RMCase) do not target the development of SWIS. Based on our knowledge the CASE tools that support the design of SWIS are now either work-in-progress (e.g., SHDM) or do say very little about how the implementation of such a system was realized (e.g., OntoWebber) in order to make these results available to the SWIS research community.

An aspect very often neglected is the ability to reuse design artifacts in building WIS. Despite the fact that there are methodologies that support the reuse of components at implementation level (e.g., AMACONT), reuse at design level is poorly sustained. The WIS design patterns in existing methodologies (e.g., WebML) usually do not consider the great potential for reuse which the Semantic Web has to offer. Also, Web services have been primarily used as a way to share data between WIS. Web services that provide a certain functionality to a WIS (e.g., data presentation or presentation adaptation) are not yet available. The existence of such services, as building blocks, would make the WIS more robust, of a better quality, and it will shorten the development time.

RDF is the foundation language for the Semantic Web. SWIS typically use RDF (or a higher-level language like OWL, also RDF-based) to represent both models and input data. This data needs to be transformed and queried in a number of subsequent steps[2]. The available RDF query languages (e.g., SeRQL, RQL, RDQL) are very often at an early development stage in which query optimization issues are not yet (or poorly) considered. It is important to note that once these languages are used for large amounts of data (as it is the case in a SWIS) the query optimization aspects are crucial for the WIS success. Also the possibility to analyze these large amounts of data can be facilitated by appropriate visualization techniques.

Hera is a SWIS design methodology. It proposes two phases: data collection, which retrieves data from different sources, and presentation generation, which produces Web presentations for the retrieved data. Hera encapsulates the best aspects from existing methodologies: the ontology-based approach from OntoWebber, the reusable components from AMACONT, the modeling style of WebML, the simplicity of RMM, etc. In addition, it focuses on the adaptation aspects involved prior or during user browsing that are considered early in the design process. A CASE tool, the Hera Presentation Generator (HPG), was developed in order to support the presentation generation phase of Hera. Several applications have been built using the HPG: a review system for the Hera papers, a shopping site for vehicles, a portal for a virtual paintings museum (without user interaction), and a shopping site for posters depicting paintings, etc. As Hera uses RDF for its specification language, significant work was also done on query optimization and data visualization of RDF data.

For all these reasons Hera is a unique answer to the complex task of SWIS design. It is based on a blend of traditional software engineering design steps with (Semantic) Web specific design steps. Each step is supported by appropriate concepts, notations, and techniques.

1.2 Research Questions and Approaches

The contribution of this dissertation is the proposal of the presentation generation phase of the Hera methodology. For this purpose several questions have to be answered:

Question 1: How to design the presentation generation for SWIS?
First we look at existing (S)WIS design methodologies identifying their main features. Then we we can ask ourselves how can one devise a SWIS design methodology that will have all these characteristics and some additional ones that we consider useful. Because we do not want to propose yet another SWIS design methodology, we would like to build upon the strong points of existing methodologies taking in consideration the facilities offered by the Semantic Web. Also one needs to explore how easily the design specifications can

[2]Note that most of the WIS methodologies use the pipeline architecture in which the input of one step is the output of the previous step.

be translated in an automatic way to software components that produce a ready-to-use SWIS. This dissertation concentrates on the front-end of SWIS design, i.e., the presentation generation.

Question 2: How can we support adaptation during the design of the presentation generation for SWIS?
One of the most important features that a SWIS needs to have is its ability to support adaptation of the hypermedia presentation prior and during user browsing. We need to consider which are the adaptation "hot-spots" inside such a methodology and how one can make the adaptation specifications work in practice. The modeling of the user interaction with the system (by means of forms) is an important adaptation aspect as it allows for a better personalization of the system according to user needs.

Question 3: What CASE tools can support the design of the presentation generation for SWIS?
The benefits of CASE tools associated to a SWIS design methodology are tremendous. It does not only simplify the designer's tasks but it will also show how one can build a SWIS using the proposed methodology. Based on the methodology steps and their associated output specifications we need to consider which are the necessary support tools and how one can build them. Having these tools applied in realizing different SWIS (possibly from different domains) will validate our proposed methodology.

Next to these major research questions, there are two other research questions that we encountered while building SWIS.

Question 4: How can one realize query optimization inside a SWIS?
SWIS usually deal with large amounts of data. As such a query optimization mechanism that will shorten the system response time (to user actions) is very important[3]. Taking in consideration the query languages used in a SWIS one needs to investigate what are the possible query optimization techniques and how they can be made available for these query languages. As we consider the Semantic Web foundation language, RDF, we will ask this question in the context of RDF query languages.

Question 5: What are suitable visualization techniques for the data used by a SWIS?
Having to deal with large amounts of data (input data or even large specification models) it is important to support the SWIS designer with techniques to get a better insight into the data properties. Visualization techniques proved to be successful in the past in realizing different software engineering objectives (e.g., reverse engineering). A legitimate question would be how one can apply existing visualization techniques in analyzing a given set of data in a SWIS. As most of the considered data representations are RDF representations, will ask this question in the context of RDF data.

[3]The user's level of patience with SWIS is usually very limited, the user moves immediately to a different SWIS if he experiences long response times.

1.3 Outline of the Dissertation

The dissertation has seven chapters. Each chapter starts with an abstract underlying the main results that are presented. The first section of each chapter provides a motivational introduction. Chapter 2, Chapter 3, and Chapter 4 do not have a related work section because Chapter 2 describes the related work for both Chapter 3 and Chapter 4. Chapter 5 and Chapter 6 have separate sections that analyze the related work and its shortcomings. After the introduction (Chapter 3 and Chapter 4) or the related work (Chapter 5 and Chapter 6) the next sections focus on the proposed solution. The last section concludes a chapter suggesting possible future work.

Chapter 2 and Chapter 3 answer research Question 1. Chapter 2 looks at existing (S)WIS design methodologies and identifies their main characteristics. Based on these characteristics and some additional ones that we consider useful we propose the Hera SWIS design methodology. Chapter 3 describes the Hera SWIS design methodology. At the core the Hera methodology there are different models that specify, based on the separation of concerns principle, different aspects involved in the design of a SWIS. The concepts involved have graphical representations and (usually) model specifications are diagrams. Each diagram has an RDF(S) representation. The focus of this dissertation is on the design of the presentation generation phase for SWIS.

Chapter 3 answers research Question 2. One of the main features of the presentation generation phase of the Hera methodology is the adaptation support for the built hypermedia presentations. The first type of adaptation that is supported is the static adaptation of the presentation based on user preferences and device capabilities. This adaptation will be performed before the user starts browsing the generated presentation. In order to better personalize a SWIS we also support a second type of adaptation, i.e., dynamic adaptation by means of forms. In this way the user is able to change the generated hypermedia presentation during the browsing session. Most of the functionality of the dynamic adaptation is given by RDF queries. Parts of Chapter 3 have been previously published in [Frasincar et al., 2001; Frasincar and Houben, 2002; Frasincar et al., 2002a; Houben et al., 2003; Vdovjak et al., 2003; Houben et al., 2004; Fiala et al., 2004].

Chapter 2 and Chapter 4 answer research Question 3. Chapter 2 recalls the existing design tools that support (S)WIS design methodologies. Chapter 4 describes the Hera Presentation Generator (HPG), a CASE tool aiming at supporting the WIS designer that uses the presentation generation phase of the Hera methodology. The HPG also produces in an automatic way a SWIS based on the designer's specifications. The chapter concentrates on how the tool supports the design steps proposed by the presentation generation phase of the Hera methodology.

Chapter 5 answers research Question 4. In order to support RDF query optimization one can define an algebra composed of a data model and a set of operators that fulfill certain equivalence laws. An example of such an RDF algebra is RAL. This algebra was developed from a database perspective in the sense that it provides similar extraction operators (with similar equivalence laws) as the ones found in relational algebra. Differently than the relational algebra RAL provides construction operators for building new data elements.

Based on the identified algebra equivalence laws, a heuristic query optimization algorithm is proposed. Chapter 5 was previously published as [Frasincar et al., 2004]. It is based on previous work published in [Frasincar et al., 2002b]. This chapter also appears in [Vdovjak, 2005].

Chapter 6 answers research Question 5. The input data and design specifications of a SWIS built with the Hera methodology are RDF data. As RDF data has a graph representation we were able to successfully apply a general purpose graph visualization tool to analyze large sets of RDF data inside a SWIS. Based on the proposed visualization techniques one can answer complex questions about this data and have an effective insight into its structure. Chapter 6 will to be published as a book chapter [Frasincar et al., 2005]. It is based on previous work published in [Telea et al., 2003].

The last chapter, Chapter 7, gives a summary of the main results and indicates some possible future research directions.

Bibliography

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):34–43. http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html.

Fiala, Z., Frasincar, F., Hinz, M., Houben, G. J., Barna, P., and Meissner, K. (2004). Engineering the presentation layer of adaptable web information systems. In *Web Engineering - 4th International Conference (ICWE 2004)*, volume 3140 of *Lecture Notes in Computer Science*, pages 459–472. Springer.

Frasincar, F. and Houben, G. J. (2002). Hypermedia presentation adaptation on the semantic web. In *Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002)*, volume 2347 of *Lecture Notes in Computer Science*, pages 133–142. Springer.

Frasincar, F., Houben, G. J., and Vdovjak, R. (2001). An rmm-based methodology for hypermedia presentation design. In *Advances in Databases and Information Systems (ADBIS 2001)*, volume 2151 of *Lecture Notes in Computer Science*, pages 323–337. Springer.

Frasincar, F., Houben, G. J., and Vdovjak, R. (2002a). Specification framework for engineering adaptive web applications. In *The Eleventh International World Wide Web Conference, WWW 2002, Web Engineering Track (WWW 2002)*. http://www2002.org/CDROM/alternate/682/index.html.

Frasincar, F., Houben, G. J., Vdovjak, R., and Barna, P. (2002b). Ral: an algebra for querying rdf. In *3rd International Conference on Web Information Systems Engineering (WISE 2002)*, pages 173–181. IEEE Computer Society.

Frasincar, F., Houben, G. J., Vdovjak, R., and Barna, P. (2004). Ral: An algebra for querying rdf. *World Wide Web Journal*, 7(1):83–109.

Frasincar, F., Telea, A., and Houben, G. J. (2005). *Visualizing the Semantic Web*, chapter 9: Adapting graph visualization techniques for the visualization of RDF data. Springer.

Houben, G. J., Barna, P., Frasincar, F., and Vdovjak, R. (2003). Hera: Development of semantic web information systems. In *Web Engineering - 3th International Conference (ICWE 2003)*, volume 2722 of *Lecture Notes in Computer Science*, pages 529–538.

Houben, G. J., Frasincar, F., Barna, P., and Vdovjak, R. (2004). Engineering the presentation layer of adaptable web information systems. In *Web Engineering - 4th International Conference (ICWE 2004)*, volume 3140 of *Lecture Notes in Computer Science*, pages 60–73. Springer.

Isakowitz, T., Bieber, M., and Vitali, F. (1998). Web information systems. *Communications of the ACM*, 41(1):78–80.

Murugesan, S., Deshpande, Y., Hansen, S., and Ginige, A. (2001). Web engineering: A new discipline for development of web-based systems. In *Web Engineering*, volume 2016 of *Lecture Notes in Computer Science*, pages 3–13. Springer.

Telea, A., Frasincar, F., and Houben, G. J. (2003). Visualisation of rdf(s)-based information. In *Seventh International Conference on Information Visualization (IV 2003)*, pages 294–299. IEEE Computer Society.

Vdovjak, R. (2005). *A Model-driven Approach for Building Distributed Ontology-based Web Applications*. PhD thesis, Eindhoven University of Technology.

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.