2003

Concpetual Design of WWW Based Information Systems

Joerg Becker
University of Munster, isjobe@wi.uni-muenster.de

Christian Brelage
University of Munster, jschbr@wi.uni-muenster.de

Michael Thygs
University of Munster, ismith@wi.uni-muenster.de

Michael Ribbert
University of Munster, ismiri@wi.uni-muenster.de

Conceptual Design of WWW-Based Information Systems

Jörg Becker, Christian Brelage, Michael Thygs, Michael Ribbert
University of Muenster
Department of Information Systems
Leonardo-Campus 3, 48147 Münster, Germany
Mail: {isjobe|ischbr|ismith|ismiri}@wi.uni-muenster.de
Phone: ++49-(0)251-8338100, Fax: ++49-(0)251-8338109

Abstract

Today, companies and their information systems are facing a very dynamic and fast changing environment. Fast changing demands require varying information in varying places. Users must find relevant information in a fast and perspicuous way. Thus, information systems both have to be flexible and clearly structured to fulfill these requirements and be able to handle the increasing amount of information efficiently. WWW-based information systems provide flexibility as they separate content, layout and navigation. However, several conceptual shortcomings prevent the Web from being a more productive and efficient resource of information today. Several technical enhancements (e.g. the semantic web) have been made in order to solve these problems. However, a theoretically proved and easy to use modeling method for WWW-based information systems is still missing. Our modeling method allows modeling these information systems on a conceptual level by abstracting from technical details. By this, meaningful navigation structures can be achieved with great clarity. In this paper we will develop this method using a meta-model based approach.

Keywords

WWW-based information system, conceptual modeling, modeling method, meta model

Introduction and Related Work

Due to many advantages (e.g. easy usability, usage by simple browsers, easy information linking), information systems using common WWW-techniques (e.g. HTML, XML, CSS) have become an essential part of information system architectures nowadays. Many companies operate intranets and websites for easy distribution of information to its employees or customers. Flexibility and, most importantly, powerful navigation structures on information are the greatest benefit of WWW-based information systems.

However, several conceptual shortcomings prevent better usability and more intense appliance of WWW-based information systems today:

- Due to the vast amounts of data on the web or in intranets, efficient information retrieval becomes increasingly difficult
- Especially in large organizations a lot of problems concerning basic terms or categories of data arise (e.g. synonyms, homonyms)
- The data on the web is usually not annotated with metadata. Thus, automated processing as well as meaningful interpretation of data is rather complicated

- Navigation structures are not consistent and are mainly defined by technical rather than semantic dependencies.

A lot of work that has been done recently at W3.ORG and in the research community aims at solving these problems. Especially the initiatives that are subsumable under the term semantic web (see Berners-Lee, Hendler and Lassila (2001) for details) are related to them. Other approaches include the resource description framework (Lassila & Swick 1999, Candan, Liu & Suvarna 2001) and web ontologies (Smith, McGuinness, Volz & Welty 2002). The inherent idea of these approaches is the annotation of data with metadata to enhance automated processing and the creation of ontologies as the foundation for linkage of data on a semantic level.

The research in this domain is mainly carried out by computer scientists who concentrate on technical feasibility and formal specifications. This paper presents an approach to model WWW-based information systems on a *conceptual* level. The conceptual design of information systems engineering usually abstracts from technical details on implementation level in order to improve comprehensibility (especially for non-technical users like business people). Furthermore, a conceptual model gives the opportunity to *document* the existing information system with a simple and easy-to-use notation and builds the basis for any CASE-concept, which allows (semi)automatic generation of information systems.

In our work, we will basically focus on intra organizational information systems, however, without limiting the applicability of our method to them. The focus on company wide information systems reduces the problems concerning synonyms and homonyms described above. Nevertheless, the conceptual design of a WWW-based information system, the definition of reasonable navigation structures as well as ontologies are still the crucial and most difficult task in designing WWW-based applications.

There are several approaches and techniques to describe web applications at the conceptual level, e. g. HDM, Dexter and WebML. All these techniques have one characteristic in common: They separate different views (or dimensions) of web applications. The Dexter Reference Model concentrates on naming conventions and was introduced by Halasz and Schwartz (1990). HDM was described by Garzotto, Paolini and Schwabe (1993). WebML was proposed by Ceri, Fraternali and Paraboschi (2000).

As stated in Brelage, Ehlers and Becker (2002), WWW-based information systems can be characterized by three dimensions (comp. figure 1). The *content* dimension describes the media itself, which is presented to the user (e. g. texts, pictures, videos). The *presentation* dimension deals with the layout and appropriate presentations of the content (e. g. layout rules to support corporate identity). Finally, the *navigation* dimension describes the structure and categories of the website. In order to create a modeling method these three dimensions have to be integrated properly.

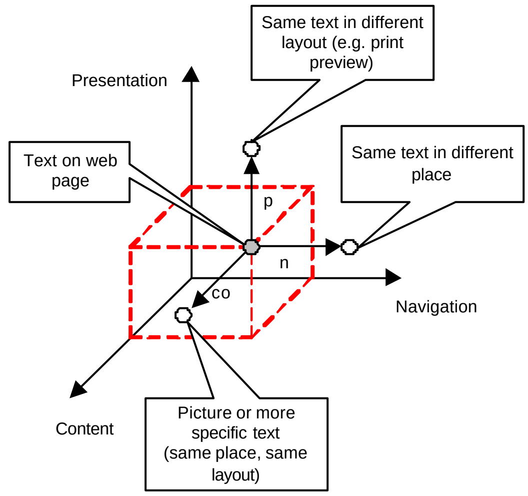

Figure 1. Dimensions of WWW-based information systems (based on Brelage et al. 2002)

The structure of this paper is as follows: In chapter 2 our research methodology will be described. Since it differs substantially from other methodologies, for instance empiricism, the basic fundamentals of method engineering will be presented in detail. In chapter 3 the ortho-language, which is the basis for our modeling method, will be described. In addition, some examples and possible notations are given to show the applicability of the method for the given domain. The last chapter contains conclusions and presents further research prospects.

Research Methodology

Information Modeling

The terms *software engineering* and *systems engineering* emphasize engineering-related methods featuring a strong theoretical foundation for developing information systems. Information models can be used as a basis for systems engineering (Karimi 1988, Kottemann & Konsynski 1984). In order to develop high quality IT solutions, business requirements need to be identified and modeled from a business perspective. After having defined the business requirements, an information system needs to be specified and can be implemented subsequently.

The Object Management Group (OMG) addresses the problem of information system engineering by proposing the so-called Model Driven Architecture (MDA) (Soley 2000). Various modeling techniques are used to develop vendor- and middleware-neutral information models. In a second step, these information models are used to design middleware concepts. After selecting a language, the implementation of information systems based on the middleware design can be initiated.

The Architecture of Integrated Information Systems (ARIS) presented by Scheer, is a further approach for specifying information systems (Scheer 2000). The four different perspectives, data, functions, organization, and control, each consisting of the three layers of conceptual model, technical model, and implementation, can be used to model different aspects of a software system from a business perspective as well as an IT perspective. All of these models correspond to each other. Language constructs from one perspective can be integrated into models with a different perspective which ensures that the information models are highly integrated. Furthermore, some perspectives support a certain degree of automation. The data model can be transformed automatically into a relational model from which a physical data warehouse schema can be derived.

An unequivocally defined taxation, a shared language based on this taxation, and shared domain knowledge between business and IT executives which is based on the shared language positively influences an improved alignment of business and IT objectives and thus enhances the quality of IT solutions (Reich & Benbasat 2000, Rosemann & Green 1999). If the business and IT staff can work collaboratively on IT specifications using the same information modeling method, it is a reasonable assumption that the requirements engineering of information systems can be simplified. In this paper we will develop a modeling method for www-based information systems.

Fundamentals of Method Engineering

According to Becker, Knackstedt, Holten, Hansmann and Neumann (2001), a method consists of one ore more techniques. Each technique contains both a language as its basic component and a direction of use as its derived component. The direction of use specifies the proper use of the language. For instance, general rules like writing or modeling from the left to the right side or principles to ensure comparability of similar language constructs are given. Ortho-language and representational aspects compound the language itself. The ortho-language provides the conceptual aspects. It gives the taxation and relationships between terms and is based on the domain the method is made for (Becker et al. 2001). Each ortho-language defines exactly one language, but one language can be represented in different ways. I.e., Entity Relationship Models represent Entity types by using rectangles, while the use of a circle can be another representation of the same semantic content (Becker et al. 2001). The following figure shows the composition of a method. As a method can hardly be tested empirically, developing a method is a constructivist approach.

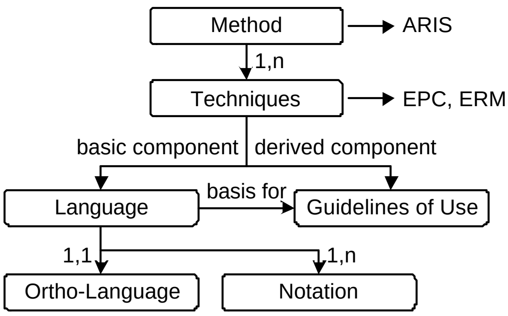

Figure 2. Composition of methods (based on Becker et al. 2001)

Meta Model Based Methods

Meta modeling is a popular approach to analyse information system methods. Based on models related to real-world objects, meta models are used to specify modeling languages (Holten 2000, Nissen, Jeusfeld & Jarke 1996, Strahringer 1996). Both the meta model and the language developed are specified by formal expressions. Modeling techniques using user adequate concepts and representations simplify the modeling process, and thus help to align further business and IT objectives (Reich & Benbasat 2000). For the development of decision support systems, a meta model based method is presented in van Hee, Somers and Voorhoeve (1991). The MetaMIS approach which integrates a meta model and a graphical representation formalism to support the specification of management views in information warehouse projects (Holten 2002) is another practically proved method, based on a thorough analysis of concepts. Whereas a model describes a real-world object itself, a meta model is usually referred to as a model of a language that describes this real-world object (Holten 2000, Nissen et al. 1996, Strahringer 1996). Thus, model and meta model are related to the same real world object. This kind of meta model is called language based meta model (Strahringer 1996). Related to the real-world object which has to be modeled, the meta model is defined in the meta language (Guarino & Welty 2002, Holten 2000).

Constructing a Modeling Method for WWW-Based Information Systems

Description of the Ortho-Language

According to the research methodology described in chapter 2, the definition of the ortho-language is the starting point of the construction of a modeling method. The ortho-language consists of fundamental terms and their relations. Each fundamental term is associated with a clear meaning. Thus, the resulting meta model (comp. figure 3), which describes the fundamental terms and their relations with a graphical notation, depicts the basic elements of the modeling method. Although the ortho-language is constructed on the meta level, instances for some fundamental terms are given for clarity reasons.

The first fundamental term is Metadata, which are used to annotate atomic elements. Each metadata may be aggregated in one ore more Metadata Sets. Metadata sets themselves may be inherited using the inheritance mechanisms known from object-oriented languages like Java. Hence, a metadata set may contain several metadata, and it is possible to construct complex metadata sets that can be used to annotate atomic elements. An example of metadata on the instance level may be <author>John Doe</author>. This annotation can be implemented using the resource description framework (see Lassila and Swick 1999 for details on RDF).

Atomic Elements represent data that build up the information system. Atomic elements are the smallest entities, which are stored in the information system. Examples of atomic elements are documents, pictures, videos or scripts (e.g. PHP or Perl) that execute database queries for dynamic content generation. Since the elements on web pages may be stored in databases, atomic elements include scripts which generate and encapsulate result sets retrieved from database queries. By encapsulating results sets, it is possible to make extensive use of metadata annotation without losing the flexibility of dynamic content generation. Moreover, atomic elements include logical entities, which are used to compile complex content. The construct of logical atomic elements is needed to add metadata sets to compilations of Content. Content is constituted by the relationship between atomic elements and metadata sets. In contrast to atomic elements (which are raw data and have no structure within themselves at all), content may be processed automatically by computers according to restrictions or descriptions given in their metadata sets. The structure Content Structure is used to build up relations between different contents.

The following example may serve to illustrate the relations between atomic elements, metadata and content: A business report consists of two chapters (atomic elements one and two), each of which is associated with a meta data set (meta data set one and two) containing information about the authors. Thus, there are two different contents. An editor wants to compile these chapters into one report and wants to add a third set of metadata (containing the expiry date for the report). In order to do so, the editor has to define a logical entity called „business report“ as well as to associates it with the third set of metadata. Hence, a third content (business report) exists and can be presented in the information system. The structure of the report is given in the relationship type content structure. The information system presents the business report and may withdraw it as soon as the expiry date is reached.

The next fundamental term of our method is Reference Object. Reference objects represent objects from the real or imaginary world. They build up a structure of terms and objects that represent important elements of the information space of an organization. Each content has to be related to at least one of these reference objects. Associating a content with a reference object means that the content object is relevant whenever information about that specific reference object should be retrieved. For instance, the business report from the example above is associated with the reference object product group=food. Thus, the content "business report" is marked to be relevant to each query that seeks information about the product group "food". The Reference Object Structure can be used to express relations between reference objects (e. g. the reference object „fruit“ is a subgroup of „food“). Reference objects may also include:

- Users and user roles to restrict content access
- Departments or assets of the organization
- Objects like regions or countries in which the organization operates

In general, everything can be a reference object. The definition of reference objects and their relations is one of the most difficult and important tasks in creating an information system. Reference objects represent ontologies and webs of important terms for an organization. It is not possible for users to navigate the information space without a clear definition of the meaning and the purpose of each reference object. Thus, it is possible to allow a meaningful interpretation of content, which is supposed to be equally meaningful to everybody who uses the information system. This concept is derived from Holten (1999) who uses a similar approach to describe the information space in information warehouse development projects. The Web Ontology Language may be used to build ontology structures on the implementation level as described by Smith et al. (2002).

The fundamental terms Navigation Node and Navigation Node Structure build up a web of nodes that may be linked to several other nodes. Each content has to be assigned to at least one navigation node. This navigation web is used to provide a free navigation without any restrictions concerning the meaning of contents. For instance, the user of the business report wants to navigate to a different part of the information system, e. g. the category "current staff" to find contact information about one of the authors of the business report. In order to provide this "shortcut", a navigation structure in addition to the reference object structure and content structure is needed.

Layout Rules and Layout Rule Sets are used to model information about the layout of content objects. These two structures can be used to create CSS-specifications (see Bos, Lie, Lilley & Jacobs (1998) for detailed information about CSS) on the instance level. In addition to this, layout rules can be used to create different representations for content objects (e. g. Braille-version for blind users).

The relationship type Content Navigation Node Assignment (Co-NN As) is redefined and associated with at least one or more reference objects. This construct allows the restriction of navigation node access via reference objects. On instance level, the triple {Business Report, Navigation Node 1, User group sales managers} means that the Business report is only viewable by sales managers at Navigation Node 1. This concept allows the partitioning of the information space by means of reference objects. The same concept applies to the relationship type Co-LRS As that allows the assignment of content objects, layout rule sets and reference objects.

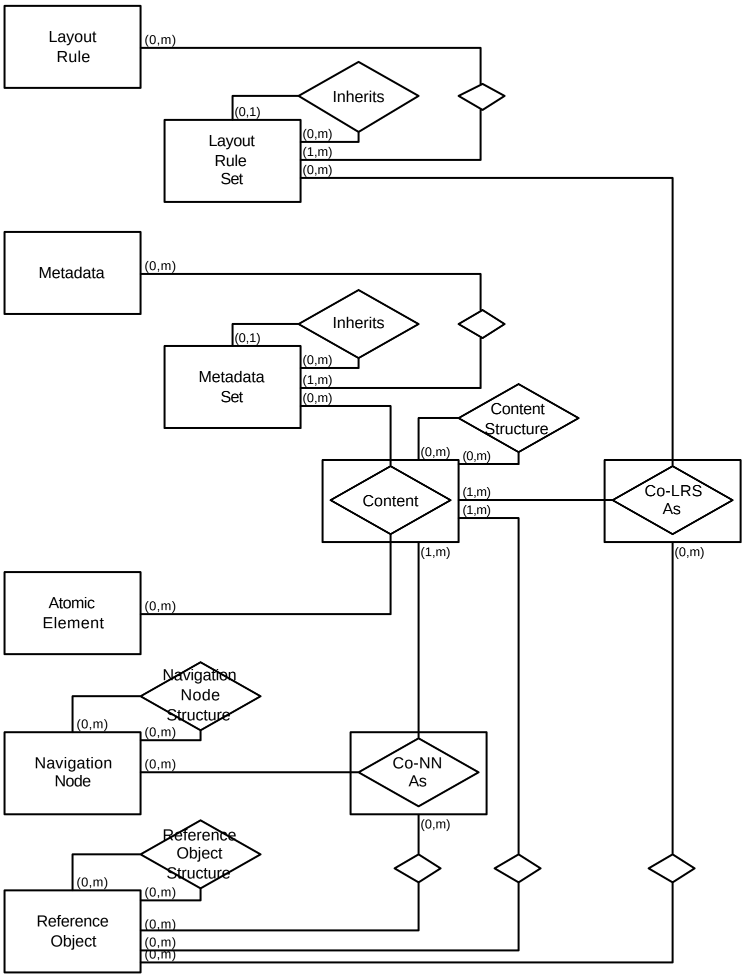

Figure 3. Meta model for the modeling method

The meta model shown in Fgure 3 is modeled using the entity relationship model known from the groundbreaking work of Chen (1976). To enhance expressiveness (min, max)-cardinalities are used (comp. Becker & Schütte 1996) instead of the common (max)-notation. Trivial relationships are not named. Note, that this model depicts the basic elements of our modeling method. It is not supposed to be directly transferred into a database schema.

Notation and Examples

As stated in chapter 2, we need a notation which is able to depict fundamental terms and their relations for our modeling method. This notation has to fulfill several requirements (e. g. easiness of use or support of efficient communication between different members of a project team) in order to improve the specification of an information system. Thus, not every aspect of our modeling method as depicted in figure 3 can or should be expressed with our notation. Several issues of our method, for instance overriding effects of metadata due to inheritance, have to be solved with algorithms on the implementation level. Moreover, it is not possible to model constraints without further enhancement of the ER-Model.

The meta model contains the most important parts of WWW-based information systems: Content, navigation and presentation dimension (comp. figure 1). The usefulness and power of WWW-based information systems is mainly constituted by the possibility to create different navigation structures. Navigation structures allow users (or robots) to switch from one content to another. Due to simplicity reasons, we will concentrate on the specification of navigation structures in the following examples and propose a simple notation to model the navigation dimension of a WWW-based information system. Note that this notation can be enhanced easily to cover the other dimensions by adding appropriate symbols based on the fundamental terms and relations given in the meta model.

First, we introduce three basic symbols of the modeling method: Content, Navigation Nodes and Reference Objects. These terms are represented with colored rectangles including an identifier in the top left corner. As described in the meta model, each of these terms can be embedded in a structure to create different types of navigation structures. The following figures show examples of different structures for the basic symbols.

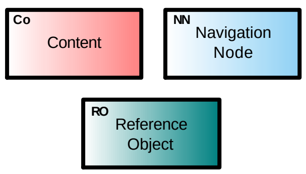

Figure 4. Basic Symbols

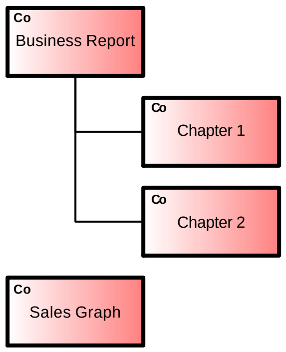

Figure 5. Sample content structure

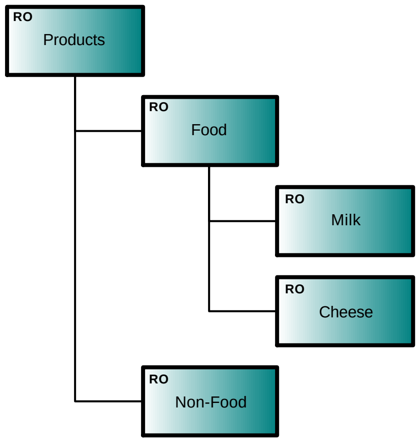

Figure 6. Sample reference object structure

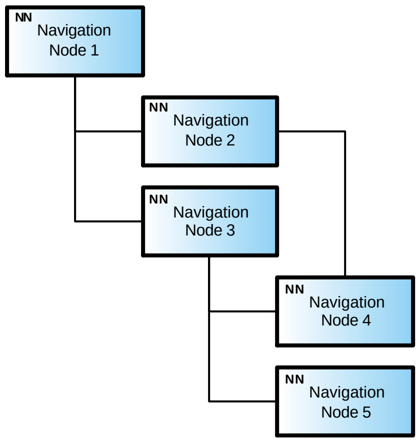

Figure 7. Sample navigation node structure

The content structure is rather straightforward and depicts dependencies between contents. Note that this example shows a hierarchy instead of a structure. Nevertheless, each content may be used at different points of the navigation web and can be related to more than one content. For instance, the dataset "contact information" is usually used at different places of the information system. To avoid redundancy and ensure consistency, this dataset should be stored in the repository only once. Therefore, a structure between content is needed rather than a hierarchy. In the example above, the business report consists of two chapters (which may be presented on different HTML-pages). The content "sales graph" is not embedded into a structure. This navigation structure can be used to navigate within complex content objects that are related to each other. This relation can be interpreted as "content A" consists of "content one and two".

As stated in the preceding chapter the construction of the reference object structure is one of the most difficult and important tasks in developing a WWW-based information system. Associating two reference objects with each other means they are related to each other semantically. Figure 6 shows a small example of a reference object structure. In this case, there is a subordination between the reference objects "food", "milk" and "cheese". The whole reference object structure forms an ontology for objects and their relations of an organization. All users of the information system are supposed to share the same interpretation of each reference object. This construct is our link to the research activities concerning the semantic web. By assigning ontology elements and metadata to web resources, the semantic web tries to improve automated processing and meaningful interpretation of data on the web. Note, that the structure of reference objects may be partitioned. Thus, it is not ensured that each node can be reached by any other node of the reference object web. Therefore, some parts may not be reached by users (this can be done by navigating via navigation nodes).

The reference object structure may be derived from predefined and common structures, that are known to the organization which implements the information system. Thus, the expense of building reference structures can be decreased and a common understanding based on existing structures can be ensured. Examples for common or predefined structures include:

- The organizational structure of an organization: Each organization can be characterized by an organizational structure that includes responsibilities and disciplinary assignments of employees and organizational units like departments or roles. Usually, this structure is modeled with an organizational chart.
- Several non-profit organizations or initiatives (for instance EAN (www.ean.org), CCG (www.ccg.de)) provide industry wide standards for article numbering and structuring of commodity groups. These predefined structures or hierarchies are the basis for any electronic data interchange between companies or organizations. They can be used to derive reference object structures, which are generally accepted even between two different organizations.
- Many organizations use several other hierarchies or structures to organize their operations. For instance: Hierarchy of regions in which the organization operates or the hierarchy of time (year, quarter, day, hour).
- The elements of the process structure of an organization may also be used as reference objects. This approach offers new opportunities of linkage between a WWW-based information system and other applications like ERP-Software: By assigning content to a certain function of an ERP-System (e.g. Order Processing), employees are instantly able to find information about that specific function (e.g. a textual description on order processing). Technically, this linkage can be realized by passing parameters from the ERP-system to the information system, which clearly identifies that function (e.g. the SAP transaction code).

Note that each reference object may be related to more than one reference object, which results in a complex structure of reference objects. However, as the examples above have shown, many reference objects are ordered hierarchically. Thus, building navigation structures on these hierarchies is even more straightforward.

The navigation node structure builds a web of nodes that can be used for navigation purposes. In contrast to the reference object structure or content structure, there is no semantic dependency or meaning implied in assigning two navigation nodes. This structure is mainly constituted by technical requirements and offers free navigation. Nevertheless, parts of this navigation space may be partitioned using reference objects (e.g. for security reasons) as described in the preceding chapter. Moreover, this structure contains links to external resources.

Up to now, we have defined three navigation structures, each of them representing a different aspect or level of navigation: A navigation about parts of content, a navigation on semantic dependencies and navigation without any restrictions. In order to create powerful navigation structures, these three views on navigation have to be integrated. This integration is shown in the figures below.

Figure 8 depicts the assignment of navigation nodes, content and reference objects. As stated in figure 2, guidelines of use are needed to use the language properly. In this case these guidelines include the following modeling rules: The symbols always appear in the same order (from left to right: navigation node, content, reference object). Lines indicate relationships that resolve to links on the implementation level. A triple “navigation node, content, and reference object” is called Information Unit. Although each content may be interpreted by readers without any reference object, the explicit assignment of reference objects is essential to ensure the same understanding by all readers. Thus, it is now possible to provide content that is explicitly associated with meaningful terms embedded in the information space of an organization.

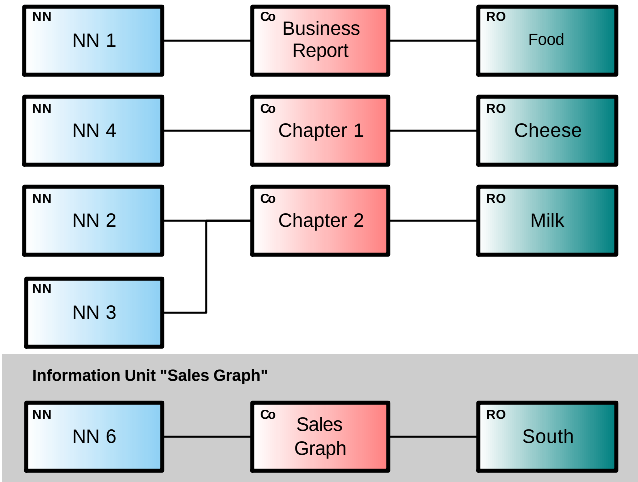

Figure 8. Assignment of content, navigation node and reference object to Information Units

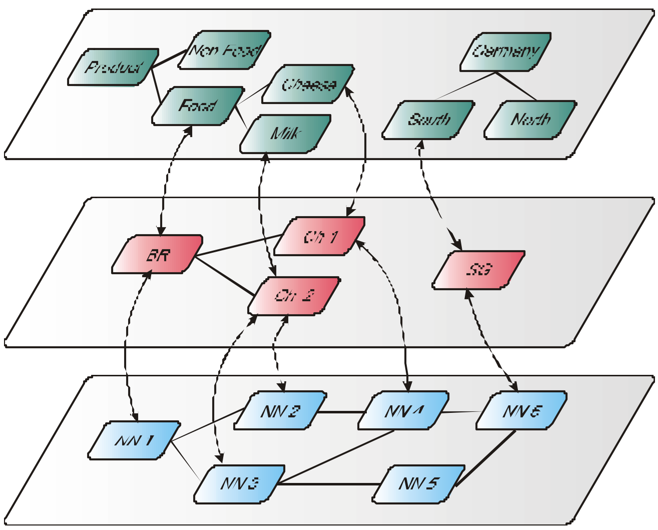

Figure 9. Navigation paths

Figure 9 shows different navigation paths. Users can navigate through the systems via related reference objects on the Reference Object Level. This way of navigation is based on certain individual defined semantic dependencies between the different reference objects. As shown in figure 9, there may be reference objects that are not semantically linked (Germany and Products). To enable users to navigate to objects, which are semantically not linked, a third navigation level is needed. The Navigation Level provides a way of navigation, which is semantically independent. On Content Level, users can navigate through the system by the structure of the content itself. As shown in figure 9 and explained above, the user is able to navigate through the chapters of the business report.

The following figure illustrates the relationship between the three different types of navigation and a possible implementation of an application which has been designed using our modeling method:

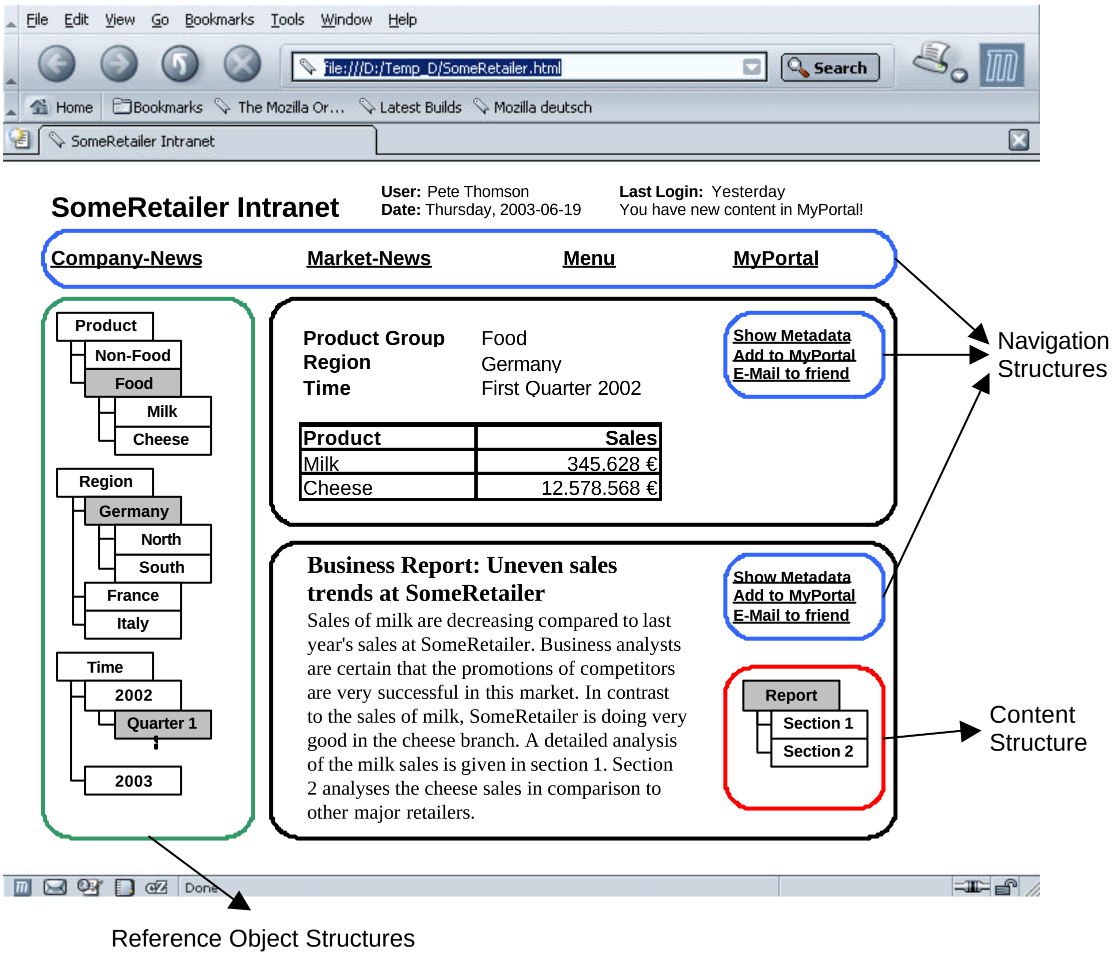

Figure 10: Example Application

The example application contains all three types of navigation explained above. Currently, the user is viewing contents that are assigned to the reference objects `Product:Food, Region:Germany` and `Time:Quarter 1` There are two contents assigned to this combination of reference objects: The current sales report and the business report. The current sales report is dynamically generated by a SQL-query that is embedded in a script which has been assigned to this combination of reference objects. The business report contains an analysis from a magazine specialized on the food retailing industry. Since this report is structured in a main part with two subordinated sections, the content structure navigation is shown in the red rectangle. The blue rectangles provide free navigation. It is important to note that although these three types of navigation are implemented by links, they differ substantially. By using the reference object structure, the user changes the semantic context given by the combination of reference objects. In contrast to this the content structure is used to organize content by means of structural dependencies. A free navigation is provided to enable users to access other parts of the information space or enhanced functions quickly.

The advantages of a WWW-based information system using our method for a conceptual design are as follows:

- Analysts are able to acquire and model the requirements for a WWW-based information system with an easy-to-use notation. Even employees who are not familiar with common web standards can contribute to the overall success of the project. Nevertheless, the model is formal enough to support CASE-driven web site development.
- Implementing well-defined navigation structures which are derived from existing structures that are commonly known in the organization enhances the usability and productivity of WWW-based information systems.
- The support of advanced web functionality like metadata annotation with the resource description or web ontologies is inherent in the model. Thus, this approach can cover future developments in this domain.

Although this approach requires a lot of work during the analysis phase of information systems engineering, it is reasonable to expect that the advantages will outbalance the disadvantages.

Conclusions and Further Work

We presented an approach to model WWW-based information systems with an easy-to-use-and-understand notation. Thus, we are able to model the semantic web on a conceptual level. By using a constructivistic research methodology, the reconstruction of our ortho-language as the basis for the modeling method, is theoretically ensured. Hence, we are able to define fundamental terms and relationships - resulting in a modeling method that is able to solve several conceptual shortcomings and inefficiencies mentioned in the introduction.

Our further research will concentrate on the enhancement of our method in order to provide a more powerful tool for information systems engineering. The modeling of the information space of an organization can be enhanced to depict more complex structures and dependencies. At the moment, our model does not use concepts like dimensions or dimension groupings which are commonly used in data warehouse design. This concept can be adapted to fit our needs. In the example application 'Region', 'Time', and 'Products' are dimensions that can be used to specify the information space more detailed. These extensions of our modeling method are currently under review.

Moreover, the guidelines of use for the modeling technique have to be improved according to figure 2 in order to use the method properly. Since the modeling of the different structures is our first approach, some problems (especially modeling more complex structures) cannot be solved satisfactorily at the moment and will have to be evaluated in detail.

We would like to thank our reviewers for their recommendations on our work. Furthermore, we would like to thank the German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung), which funded this work under record no. 01HW0196.

References

Becker, J & Schütte, R (1996), *Handelsinformationssysteme*, verlag moderne industrie, Landsberg, Lech, pp. 75-82. [in German]

Becker, J, Knackstedt, R, Holten, R, Hansmann, H & Neumann, S (2001), 'Konstruktion von Methodiken: Vorschläge für eine begriffliche Grundlegung und domänenspezifische Anwendungsbeispiele' in *Working Papers of the Institute of Information Systems*, no. 77, Muenster. [In German]

Berners-Lee, T, Hendler, J, & Lassila, O (2001), 'The Semantic Web', viewed 31 March 2003, <http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>.

Bos, B, Lie, HW, Lilley, C & Jacobs, I (1998), 'Cascading Style Sheets Level 2 Spezification', viewed 31 March 2003, <http://www.w3.org/TR/REC-CSS2/>.

Brelage, C, Ehlers, L & Becker, J (2002), 'Modeling and implementing macro web navigation structures' in *Proceedings of the Xth European Conference on Information Systems*, p. 1547, Poland, Gdánsk.

Candan, KS, Liu, H, & Suvarna, R (2001), 'Resource description framework: metadata and its applications', *ACM SIGKDD Explorations Newsletter*, vol. 3, no. 1, 2001.

Ceri, S, Fraternali, P & Paraboschi, S (2000), 'Web Modeling Language (WebML): a modeling language for designing Web sites' in *Proceedings of the 9th International World Wide Web Conference*, Elsevier, pp. 137-157.

Chen, PPS (1976), 'The Entity-Relationship Model - Toward a Unified View of Data', *ACM Transactions on Database Systems*, vol. 1, no. 1, pp. 9-36.

Garzotto, F, Paolini, P & Schwabe, D (1993), 'HDM - An model-based approach to hypertext application design', *ACM Transactions of Information Systems*, vol. 11, pp. 1- 26.

Guarino, N & Welty, C (2002), 'Evaluating Ontological Decisions with OntoClean', *Communications of the ACM*, vol. 45, no. 2, pp. 61-65.

Halasz, F & Schwartz, M (1990), 'The Dexter Hypertext Reference Model' in *Proceedings of the Hypertext Workshop*, National Institute of Standards (NIST) Special Publication, Gaithersburg, pp. 95-133.

Holten, R (1999), *Entwicklung von Führungsinformationssystemen. Ein methodenorientierter Ansatz*, Gabler, Wiesbaden. [in German]

Holten, R (2000), 'Framework and Method for Information Warehouse Development Processes' in *Proceedings of the Data Warehousing 2000 Conference*, Physica, Heidelberg, p. 135.

Holten, R (2002), 'Specification of Management Views in Information Warehouse Projects' in *Working Papers of the Institute of Information Systems*, no. 87, Muenster.

Karimi, J (1988), 'Strategic Planning for Information Systems: Requirements Engineering and Information Engineering Methods', *Journal of Management Information Systems*, vol. 4, no. 4, pp. 5-24.

Kottemann, JE & Konsynski, BR (1984), 'Information Systems Planning and Development: Strategic Postures and Methodologies', *Journal of Information Systems*, vol.1, no. 2, pp. 45-63.

Lassila, O & Swick, RR (1999), 'Resource Description Framework (RDF) Model and Syntax Specification', viewed 31 March 2003, <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>.

Nissen, HW, Jeusfeld, MA & Jarke, M (1996), 'Managing Multiple Requirements Perspectives with Metamodels', *IEEE Software*, vol. 13, no. 2, pp. 37-48.

Reich, BH & Benbasat, I (2000), 'Factors That Influence the Social Dimension of Alignment Between Business and Information Technology Objectives', *MIS Quarterly*, vol. 24, no. 1, pp. 81-114.

Rosemann, M & Green, P (1999),' Enhancing the Process of Ontological Analysis – The "Who cares" Dimension"' in *Proceedings of the Information Systems Foundations Workshop – Ontology, Semiotics and Practice*, Lighthouse Press, Sydney, p. 39.

Scheer, A-W (2000), *ARIS - Business process Modeling*, 3rd edn, Springer, Berlin.

Smith, MK, McGuinness, D, Volz, R & Welty, C (2002), 'Web Ontology Language (OWL) Guide Version 1.0', viewed 31 March 2003, <http://www.w3.org/TR/owl-guide/>.

Soley, R (2000), *Model Driven Architecture,* White Paper, Needham.

Strahringer, S (1996), *Metamodellierung als Instrument des Methodenvergleichs. Eine Evaluierung am Beispiel objektorientierter Analysemethoden*, Shaker, Aachen. [in German]

vanHee, KM, Somers, LJ & Voorhoeve, M (1991), 'A modeling environment for decision support systems', *Decision Support Systems*, vol. 7, no. 3, pp. 241-251.