From Single-Site Web Applications to the Design of Web-Site-Families

Christian Eichinger, Michael Schrefl *

Abstract

Web applications, or web sites, are an important communication medium conveying information between organizations and the users of their web applications. To ensure a unique appearance of their web applications, organizations define standardized content, navigation and presentation requirements for web applications within their corporate identity. However, specialized information supplied by organizational units may not fit into this standardized scheme. Web site families try to overcome this situation by capturing the requirements necessary to fulfill the corporate identity, while at the same time providing the flexibility to specify requirements which are particular for individual organizational units. For this purpose, they rely on a hierarchical model in which common requirements for a family of web sites are captured. Those requirements may be extended, refined or re-structured within a concrete web site of an organizational unit as long as it adheres to the common requirements. In this paper the structure of web site families as well as consistency criteria that allow the flexible definition of families of web sites, are presented.

Keywords: conceptual design, hypertext/hypermedia, consistency, design guidelines, architecture

1 Introduction

Organizations use a wide variety of web applications to share information via the Internet, ranging from simple, active HTML pages to fully fledged software systems utilizing back end business processes. The development of such web applications is a complex activity requiring appropriate methods and tools [8]. Especially for the development of data intensive web applications and for the design of hypertext a wide range of methods (Araneus [11], OOHDM [15], OOWS [12], Strudel [7], WebML [3], WSDM [17], etc.) and accompanying tools (e.g. the WebRatio tool for WebML) have been proposed.

Most of these design methods allow to define content, navigation and presentation models to specify a web application. However, a developed web application may not be connected or related to other web applications of the organization and thus the current design methods do not allow to comfortably coordinate the design of related, similar web sites.

Especially large or distributed organizations have to capture a wide variety of requirements of different organizational units. Universities are examples for such organizations where faculties and departments of different research areas expect different kinds of content, navigation and presentation from a web application. To accommodate all these different requirements into a single web application model is very difficult at best. In addition, information shared between the organizational units, the departments, should be taken into account as well. Furthermore, the information provided via a web application must adhere to the corporate identity of the organization. At present, such organizations may choose between the following two extremes:

1. The organization standardizes the content, navigation and presentation requirements for the entire organization. Typically, these requirements are captured within the corporate identity of the organization. The fulfillment of the corporate identity is enforced by the usage of centralized tools into which the organizational units feed their information.
2. The organization allows each organizational unit to specify its content, navigation and presentation with respect to their specific needs. The resulting web applications are designed without any common guidelines, thus sharing only little commonalities.

These choices have advantages and disadvantages for both the users and administrators of web applications. Users of standardized web applications will experience a consistent browsing behavior as content, navigation and presentation designs are identical for all web applications of the organization. However, information specific to organizational units will not be available from the official site. This kind of information typically is swapped to external web sites which are out of the reach of organizational regulations (e.g. at personal web pages of staff). Administrators, on the other hand, only have to administrate a single web application, thus saving time and costs.

Users of independently administrated web applications will get the most specific information available, presented

*Johannes-Kepler-University Linz, Department of Data & Knowledge Engineering, Altenbergerstr. 69, 4040 Linz, Austria. Email: eiching@dke.uni-linz.ac.at. A previous, shorter version of this paper was originally published in the Proceedings of the IASTED International Conference on Software Engineering 2006.

in the most appropriate manner. However, they will need to get accustomed to a variety of different web applications and thus the cognitive overhead for them is high. In addition administrators will have to manage a wide variety of different web applications and tools used to generate them. As a result, independent web applications render the enforcement of, e.g., a common security policies or a single user administration impossible.

To solve these problems, decentralized organizations capture their corporate identity within organizational rules to which the web application designers must adhere. As a result, a higher administrative overhead is traded in for additional flexibility. However, misinterpretation of rules as well as evolving rules require the constant re-design of web applications which often leads to inconsistencies between web applications and the corporate identity. In addition, the correct implementation of corporate rules has to be monitored continuously to ensure that single web applications do not drift off from the corporate identity.

To overcome these problems, we propose the use of a hierarchical web application model, so called web site families, in order to provide a technical support for the organizational solution described above. Rules collecting common aspects of web applications as well as specific requirements of different organizational units are captured within web site families. Based on these rules, a technical system for enforcing the corporate identity can be built.

In this paper, we will limit ourselves to describing the design principle behind web site families as well as the consistency criteria allowing the development of web site families based on common content, navigation and presentation models. The challenges, web site families face with respect to these requirements as well as consistency rules deduced from these requirements are detailed in Section 2. The consistency criteria for realizing web site families on top of web application models are detailed in Section 3. Section 4 reviews the related work. Finally, Section 5 concludes the paper and summarizes open research questions.

2 An overview of web site families

Web site families are defined over a set of similar web applications or web sites - their members. Looking at the university department of Software Engineering (SWE) and the department of Data & Knowledge Engineering (DKE), visualized in Figure 3, we can identify similar content, navigation and presentation aspects.

The notation used in the examples is borrowed from existing approaches. At the content level, we use a class model similar to that of the UML [2]. However, the notation is extended with a role relationship, visualized by a dashed line with arrow as, e.g., used between the department classes in Figure 2. In addition, a dashed association is introduced, that represents an association that is non-invocable. Invocability is described in Section 3. Derived attributes are represented by a trailing slash as, e.g., at role class department in Figure 2. The notation used at the navigation layer is borrowed from the WebML [3] notation. Units are visualized as rectangles and combine the functionality of index- and multidata- units specified in WebML. Links are visualized by arrows and pages as dotted rectangles. For the examples used in this paper we do not need a more sophisticated hypertext model and thus we omit elements like parameterized links. The presentation model borrows its ideas from screen description formats that are part of e.g. the SMIL [6] language. We use screen descriptors (represented by a dashed and dotted line) for grouping all regions (represented by solid rectangles) of a screen. Sub-regions are specified by dashed lines separating a region in multiple sub-regions. A page or unit is associated to each region/sub-region for display, indicated by the name of the region/sub-region. Otherwise, all dashed lines frame undefined regions, i.e. display area on screen that is, so far, unused and may be used for defining further regions (cf. Figure 2).

At the content layer the DKE and SWE department capture information about staff and the fields of research. At the navigation layer they provide access to the fields of research, and at the presentation they display header information and a navigation menu. However, they also exhibit distinct aspects within their models, like the content classes “news” or “projects”, the different navigational paths to the “fields of research” or the different positioning of the menu of the web sites.

However, as long as the similarities and differences of web pages are not made explicit, it is extremely difficult for users, e.g. students, to access the information they are interested in. Although the desired information may be shared among multiple web applications, different layout, navigation or presentation may render it difficult for users to access, identify and process information. As a result, the cognitive overhead for accessing the information is rather high. However, not only end users are affected.

Administrators will find it very difficult to maintain or enforce common requirements upon diverse web applications, too. Usually, they lack adequate tools for capturing and enforcing corporate identity rules upon such web applications. As a result, they are burdened with the task to manually interpret and enforce organizational rules documented in the corporate identity.

To overcome these problems, we propose a hierarchical model of web site families to explicitly capture similarities and differences between web applications. Each level of a web site family hierarchy composes a web site *family* subsuming all web applications at this level. A web application in a family is referred to as *member* of this family. The hierarchy defined over web site families is reflected at the member level as well, i.e. each member of a family is connected to members of the hierarchically lower family. The hierarchical relationship defined by family members is referred to as *member hierarchy*.

In the example shown in Figure 1, the left hand side shows the web site family hierarchy of the example and the right hand side its corresponding member hierarchy. The lowest level of the sample hierarchy defines the family of departments. All department web applications (DKE, SWE, physics, chemistry) are members of the department family. The member hierarchy is rooted at the University of Linz which refers to its faculties, the faculty of social sciences and the faculty of technical sciences, which in turn refer to their departments.

Each family is divided into one or more sub-families. Sub-families combine those members of a family, that share similar content, navigation or presentation aspects. Each member of a family is associated with exactly one sub-family. Sub-families are formed along the member hierarchy, i.e. only those family members may be combined into a sub-family, that refer to the same parent. The direct parent of all members of a sub-family is referred to as the sub-family parent. The similarities of the members of a sub-family are captured in a *sub-family description* which is associated with the sub-family parent.

The dotted rectangles in Figure 1 show the sub-families of our example. The sub-family at university level is equal to the family itself as it only contains the University of Linz. For the faculty family, one sub-family containing the two family members is defined. At the department level, three sub-families are defined: one for the SWE and DKE departments (described at the faculty of social sciences), one for the physics and chemistry departments (described at the faculty of technical sciences), and one for all departments, specified at the university level.

For the remainder of the paper we will use the following structure of sub-family descriptions. The DKE and SWE departments are associated with the faculty of social sciences, the physics and chemistry department are associated with the faculty of technical sciences. The sub-family description for the department family as a whole is defined at the University of Linz and is passed on to the department sub-family descriptors of its faculties. In addition, the University of Linz defines the sub-family description for its faculties. As a result, the University of Linz captures requirements for both, its faculties and its departments. The faculties may only specify sub-family descriptions for their departments, taking into account the definitions from the University of Linz.

The sub-family descriptions are visualized by black squares in Figure 1, that refer to the sub-family they describe. A more complex sub-family description model would allow to specify arbitrary sub-families allowing to define sub-family descriptions at arbitrary hierarchy levels, e.g. the university could specify distinct requirements for the physics and chemistry department. However, a more complex sub-family description model is not within the scope of this paper and thus we will stick to the simple model explained previously.

The extreme choices described in Section 1 fit into this

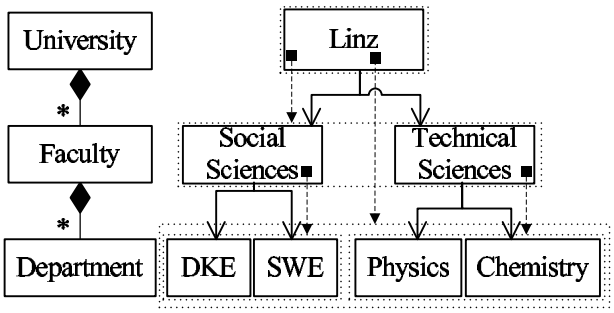

Figure 1: Example hierarchy between web site families

hierarchy in the following ways. The centralized solution would be represented as a single sub-family description at the University of Linz, capturing the entire department model. Thus, the entire model would be compulsory for all departments, leaving them with no design choices. The second extreme of totally design freedom for each department, could be realized by not defining a single sub-family description. Thus, any web application model defined at the department family would be valid.

In order to capture similarities and particularities of web site families, a family member has to capture two kinds of web application models. The first kind specifies a *local* model for the content, navigation and presentation aspects of the family member itself. The second kind comprises of web application models for each child family, captured by the respective sub family description. The local model of a family member thus allows to capture the particularities of the family member while the sub-family description defines common requirements for the children of this member.

For example, Figure 2 describes the faculty of social sciences, a member of the faculty family. The left part of the family describes content, navigation and presentation aspects of the faculty itself. The right part of the model represents the sub-family description of its department sub-family. The sub-family description, e.g. define "staff" and "research fields" as compulsory parts of the content model. Both departments fulfill this requirement, as the staff and research field classes are available in their local models, marked in grey (cf. Figure 3).

However, to automatically evaluate, whether a web application model of a family member is consistent with its sub-family description, we first need to identify how a web site model might be specialized. Table 1 summarizes different specializations types at the content, navigation and presentation layer of web application models. The examples given refer to Figure 3.

Now, that we know about possible deviations within web application models, we are able to enforce sub-family descriptions by defining consistency criteria over web application models. Those criteria must allow us to evaluate, whether a web site model of a sub-family member is a consistent extension, refinement or re-structuring of a

Layer	Type	Description
Content	Extension	Web applications may differ in the amount of information captured, e.g. the DKE department keeps information about the latest news, relevant for students of the department, while the SWE department does not directly record this kind of information.
	Refinement	Web applications may store information in different levels of detail, e.g. the DKE department differentiates between scientific and non scientific staff members while the SWE department does not.
	Re-structuring	Web applications may structure information differently, e.g. at the SWE department research fields are associated to projects while at the DKE department this information is related to the scientific staff members.
Navigation	Augmentation	Web applications may differ in the number of links and pages available for navigation, e.g. at the SWE department, the project information extended at the content layer needs to be represented in the navigation model as well.
	Expansion	Navigation to a certain piece of information may be detailed into multiple steps, e.g. staff information at the DKE department is reachable in one step, while at the SWE department, staff information may be accessed via the projects page.
	Composition	Information may be scattered on multiple connected pages or may be available on a single page, e.g. the SWE department has different web pages for each research field, while the DKE department uses a single page to display staff as well as research field information.
Presentation	Addition	The number of display regions increases as additional information is displayed, e.g. beside the DKE department header the university logo is added.
	Partitioning	A display region is partitioned in order to detail existing information, e.g. the staff and research page of the DKE department is partitioned into three subregions accommodating information on non scientific staff, scientific staff and the fields of research of the scientific staff.
	Positioning	The actual positioning of regions on screen may vary, e.g. the navigation menu of the DKE department is located at the left handside of the screen while the SWE department positioned the navigation menu on the top right part of the screen.

Table 1: Types of specialization of web sites, structured according to web application layers

sub-family description. In Section 3 we give an overview of the consistency criteria needed to realize web site families.

3 Consistency criteria for web site families

Consistency criteria over web site families guarantee, that the common aspects defined in sub-family descriptions are met by their respective members. These criteria assure the similarities intended between web applications, making it easier for users or administrators of web applications to use and administrate them. With consistency criteria met, users and administrators will recognize familiar content, navigation and presentation aspects within the web application models of all sub-family members.

Organizational units like departments, should, however, still be able to adapt their web applications to their specific needs. They should be able to extend, refine and re-structure their web application as far as possible in order to provide this flexibility. A specialization of sub-family descriptions should therefore still be possible.

The consistency criteria for web site families thus have to balance corporate identity guidelines against the required degree of flexibility. As a result, we distinguish requirements that must be fulfilled unchanged and those that may be subject to specialization. The minimal requirements these rules should guarantee are defined by the needs of users and administrators, which can be summarized with the following points:

- *Transparency* allows users to change hierarchies in a transparent manner, i.e. they may follow refinements without explicitly performing a hierarchy switch. For administrators, transparency is necessary to pass on changes down the member hierarchy of web site families.
- *Awareness* allows users to orient themselves in the web application model of a sub-family member according to its parent sub-family description. The sub-family description acts as an orientation guide for the user. For administrators, awareness is necessary to estimate the implications that changes entail on a model.
- *Recognizability* allows users to relate content, navigation or presentation aspects of a sub-family member

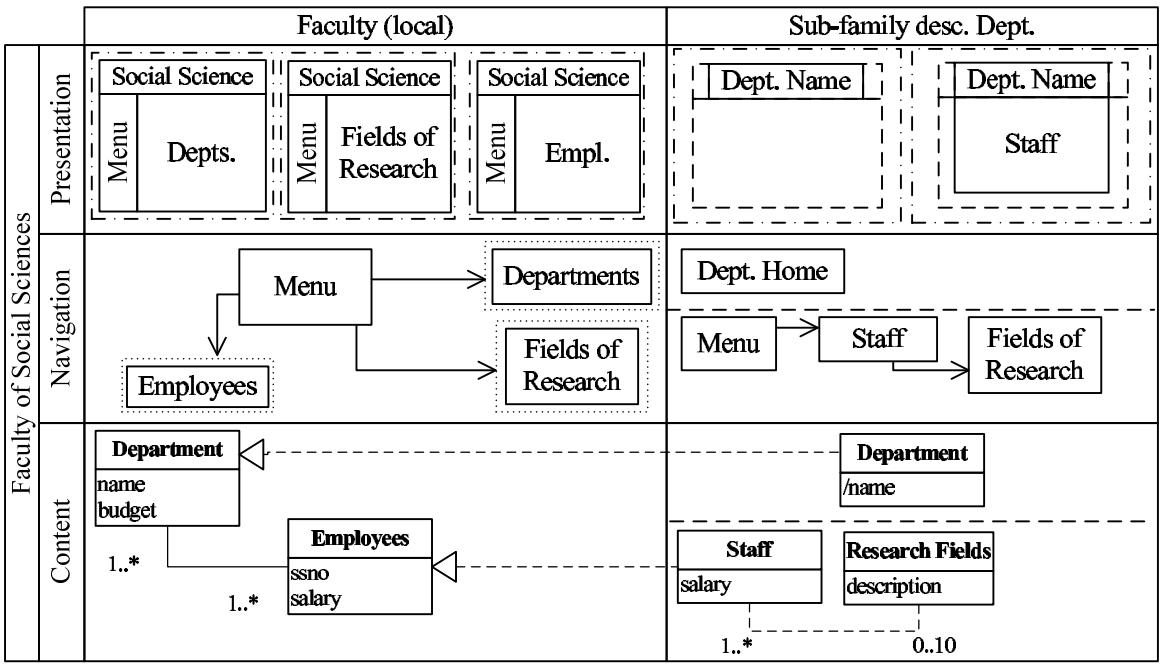

Figure 2: Example model for the faculty of social sciences

model to the respective aspects within the sub-family description.

Based on these requirements we can define two general consistency criteria that have to be fulfilled on content, navigation and presentation layers of web applications:

- *Observability* guarantees, that any operation performed at a sub-family member model is observable at its sub-family description and yields a valid result. A result at the sub-family description is valid, when it conforms with the result defined at the sub-family description, i.e. either it is an equal or a refined result. Operations can be the access, traversal between or the changing of information.
- *Invocability* guarantees, that an operation performed according to the sub-family description may also be performed at any sub-family member model yielding a valid result. Operations and valid results are defined as above.

As we can see, observability provides a bottom up consistency between sub-family members and sub-family descriptions and invocability provides a top down consistency between them. Following these consistency criteria we can ensure, that web application models of sub-family members extend, refine or re-structure sub-family descriptions in a well defined, consistent way.

In the following subsections we will concretize observability and invocability for each web application layer, exemplified with the models shown in Figure 2 and Figure 3. Information specified in the local model of Figure 2 may be related to information at the sub-family description. The role relationship between the classes employee and staff specifies, that staff members of a department are referred to as employees at the faculty and thus, all staff members of a department are also employees of the faculty. The other role relationships can be interpreted analogously.

In addition, the faculty of social sciences captures information about its departments (name and budget). This information is passed on to the departments as meta-data about themselves. To distinguish this meta-data from the rest of the information provided, a dashed line is used to split the layers of the sub-family descriptions where the upper part describes the meta-data and the lower part describes the similarities required by the faculty. Only data relevant for the respective department is passed on as meta-data.

This separation is necessary, as department information usually is not explicitly modelled in a department model. Without this mechanism inconsistencies between information stored at the faculty level and that used at the department level might occur, e.g. when the department name used at the faculty level differs from that hard coded in a department HTML page.

Unfortunately, we are not able to detail all consistency criteria applicable at the different layers within this paper, as it is limited in size. The formal model dealing with these aspects exhaustively will be published in future work of the authors. However, the idea of the approach and its effectiveness should be conceivable with the provided examples.

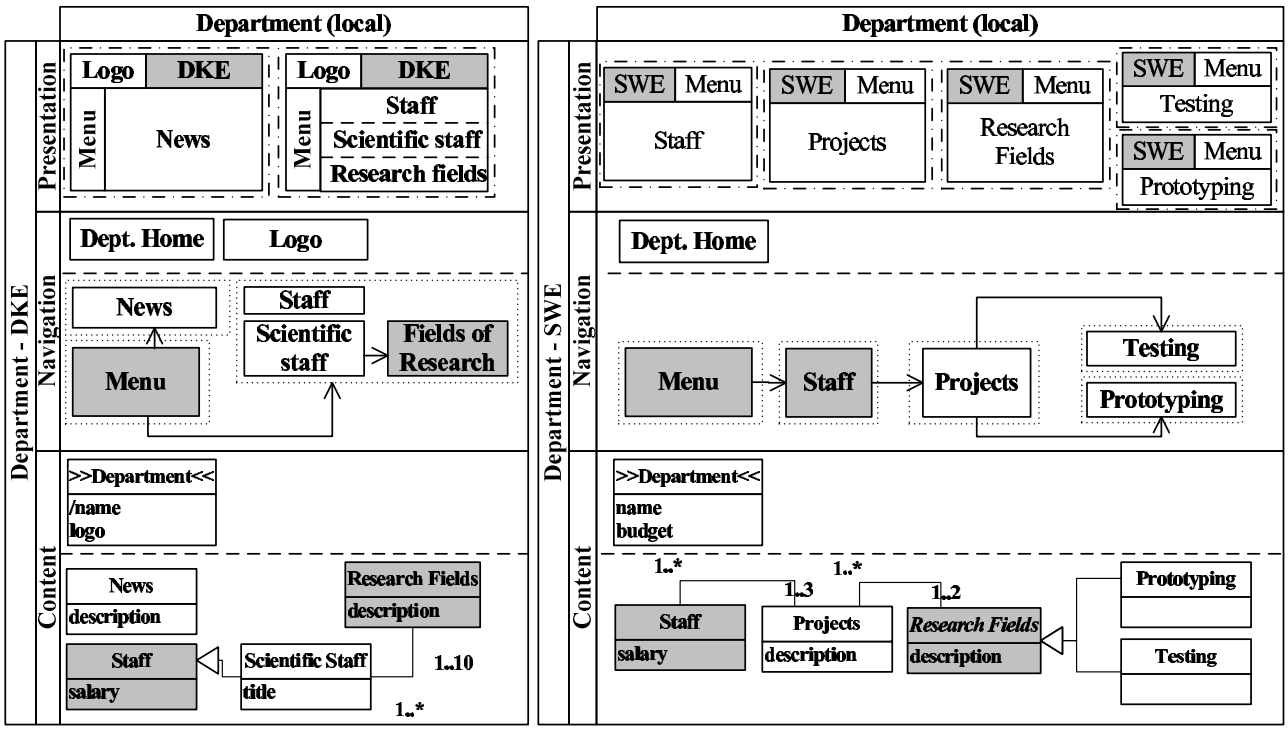

Figure 3: Example model for departments DKE (left) and SWE (right)

3.1 Content Layer Consistency

Observability and invocability at the content layer very much conform to the general definitions. Observability is fulfilled, when any access to a class, attribute, or association and any insertion, update or deletion of content that is valid on a sub-family member model may be performed as well at the sub-family description without violating any constraints such as multiplicities of associations.

Extension and refinement of content layer models affect classes, associations and attributes of classes. Thus, extending the content model adds new information to it. Adding new classes to the model typically entails an extension of associations as well, since newly added classes need to be connected to existing classes. In our examples the local models of the DKE and SWE department extend the sub-family description of the faculty of social sciences with the classes news (DKE) and projects (SWE), together with the appropriate associations. In addition, existing classes may be extended by new attributes as, e.g., the attributes budget or logo extending the department meta-data class at the local models of the SWE and DKE department, respectively.

To judge, wether these extensions are observable and/or invocable we need to look at the sub-family description of the faculty of social sciences. The extensions of the model with the classes news and project together with the respective associations do not have any correspondence in the sub-family description and thus, observability does not apply and, hence, is not violated for these specializations. The extensions of the department class with attributes logo (DKE) and budget (SWE) do not conflict with observability either.

The refinements at the DKE department, the generalization of scientific staff to staff and the association between scientific staff and research fields fulfill observability. The generalization is observable, as any access, change or insert of scientific staff members is traceable as an operation on the class staff at the sub-family description. The refined association fulfills the criteria of observability, as a traversal from scientific staff members to their research fields can be observed as a traversal of staff members to research fields at the sub-family description. Changes of this association (insert, update or delete) may also be observed as respective changes on the staff to research fields association. The inverse traversal, analogously, is fulfilled as well. The multiplicities defined at the association in the sub-family description allow for staff members not to be associated with a research field and thus covers the case of staff members not being scientific staff at the DKE department model.

The refinements at the SWE department, the subclassing of the research fields and the association of staff members to research fields via the projects class, fulfill observability. A traversal from staff via projects to research fields can be observed as a traversal from staff to research fields at the sub-family description. The same is true for a traversal from research fields via projects to staff. Any change of a project, can be traced as change of the staff to research fields association as well. The multiplicities defined for the associations in the SWE model also fulfill observability, as via projects any research field must have

at least one staff member associated, and each staff member is associated to at least one and at most six research fields.

As none of the department models violates observability the sub-family description is fulfilled with respect to observability. As a result, any content access or traversal of associations applicable, can be mapped onto the sub-family description.

Invocability on the other hand is not fulfilled. The problem originates at the association between the staff and research fields classes at the sub-family description. Although this association may be consistently traversed by reading all projects of a staff member and thus computing the associated research fields, the same does not hold for changes of the association. It is not possible to, e.g., add a new research field to a staff member as defined in the sub-family description, as at the SWE model the appropriate project information for completing the operation is needed. At the DKE department model the refinement fulfills invocability with respect to the traversal of the association, however, inserting a new association as specified by the sub-family description is not possible, as it can not be ensured, that the inserts will only take place for scientific staff members. As a result, not all operations, content access, update or traversal possible at the sub-family description can be propagated to the department models. Therefore, administrators may not perform content updates based upon the sub-family description.

3.2 Navigation Layer Consistency

Observability at the navigation layer is fulfilled, when any navigation performed at the sub-family member model may be traced at the sub-family description. The sub-family description acts as "orientation map", where the current position in the hypertext model is tagged. Any link traversal thus, must result in a target, valid at the sub-family description. Invocability is fulfilled, when a link traversal performed at the sub-family description may also be performed at the sub-family member model, yielding a valid result.

Augmentation, expansion and composition of navigation layer models affect units, links and pages. Augmenting the navigation model with new units typically entails an augmentation with links as well, as newly added units need to be connected to the existing navigation structure. In our examples the local models of the DKE and SWE department augment or expand the sub-family description of the faculty of social sciences with the pages/units news, logo (both DKE) and projects (SWE), together with the appropriate links. The augmentation of the news page/unit at the DKE department model is not observable at the sub-family description, as a link traversal to the news page at the department model yields a result (news) not specified at the sub-family description.

The expansions at the DKE department, the splitting of the staff unit into a non-scientific and a scientific staff unit and the linking of scientific staff with the field of research unit together with its composition into a combined staff and research field page, are observable. The expansion fulfills observability, as a link traversal from menu to the combined staff and fields of research page can be traced at the sub-family description as a traversal from menu to staff. The traversal of the link between scientific staff and fields of research may be traced at the sub-family description as a traversal from staff to fields of research. The composition is observable, as no explicit page structure for the units fields of research and staff is specified at the sub-family description.

The expansions at the SWE department fulfill observability, as a traversal from menu to staff is directly observable at the sub-family description and a traversal from the project to the testing or prototyping unit/page can be traced as a traversal from staff to fields of research. The composition is observable, as no explicit page structure for fields of research is specified at the sub-family description.

Invocability is not fulfilled within the SWE department model for the traversal of the link from staff to research fields as it is ambiguous, because the target of the navigation at the sub-family description (either testing or prototyping) is not known. At the DKE model, a traversal from menu to staff is invocable. However, a traversal from staff to fields of research is not always possible, as non-scientific staff members do not link to the fields of research and thus, invocability is violated.

3.3 Presentation Layer Consistency

Observability at the presentation layer is fulfilled, as long as a region or sub-region specified at the sub-family description may be added into a screen descriptor specified at the sub-family member model without overlapping another region. As a result, observability ensures that any region defined at a sub-family member model may actually be displayed within the screen descriptors specified at the sub-family description. Invocability is fulfilled, when every screen descriptor specified at the sub-family description is implemented in the sub-family member model. This ensures, that any screen descriptor specified at the sub-family description, actually will appear in the sub-family member.

Addition, partitioning and positioning of presentation layer models affect display regions and sub-regions. The augmentation of navigation models usually require the addition of new display regions at the presentation layer, as newly added information needs to be displayed on screen. In our examples the local models of the DKE and SWE department use newly added regions to display the units logo, menu, etc. besides the region for displaying the department's name.

To judge, wether these additions are observable and/or invocable we need to look at the sub-family description of the faculty of social sciences. The sub-family descrip-

tion specifies two screen descriptors. The first consists of a single region displaying the name of the department at the top of the screen. It defines, that any new screen descriptor specified at department level, must at least hold a region for displaying the department name at its top. The second screen descriptor also specifies the name region. In addition, it requires that staff information must be displayed directly below the name region. As a result, the additions defined at the presentation models of the DKE and SWE departments fulfill observability as any region specified at department level fits into a screen descriptor of the sub-family description.

The partitioning of the staff and fields of research page into multiple sub-regions at the DKE department is observable, as the partitioning still fits into the first screen descriptor of the sub-family description. However, the partitioning is not invocable, since the display of staff, as specified in the second screen descriptor, is not available.

The positioning of the name region at the right margin of the screen defined at the DKE department fulfills observability as the region at the right of the name region is undefined in the sub-family description. Thus, the name region may be re-positioned as long as it is aligned at the top of the screen. In addition, this positioning is invocable, as it is valid in all screen descriptors of the sub-family description.

The positioning defined at the SWE department, is very similar to the positioning previously described. The only difference is, that the name region is aligned at the left margin of the screen. Due to that, the positioning as specified at the SWE is observable as well.

As none of the department models violates observability at the presentation level, the sub-family description is fulfilled with respect to observability. As a result, any presentation specified at the departments, can be mapped onto the sub-family description. Invocability on the other hand is not fulfilled by the sub-family member model of the DKE department as it is not clear which sub-region (staff, scientific staff or research fields) needs to be activated when invoking staff at the sub-family description.

4 Related work

Various approaches are dealing with the refinement, extension or re-structuring of content, navigation and presentation models. At the content layer object oriented models distinguish refinement for object types, relationships and object behavior. Aiguier ([1]) for example investigates the refinement of objects into systems of object types within the algebraic formalism of Étoile. Costal et al. ([4]) specify the refinement of relationships in conceptual models with multiple classification and Schrefl and Stumptner ([14]) for example specify the refinement of object behavior.

The augmentation, expansion and composition of navigation structures are not as thoroughly investigated as object oriented refinement, extension and re-structuring. OOHDM and its extensions ([16, 13, 10]) specify a navigational meta-model which allows to parameterize navigational links, enabling the hypertext designer to instantiate them in a web application. The Content Aggregation approach of OOWS ([18]) allows the hierarchical specification of navigational units using information abstraction units. Statecharts are used in [19, 9] as a more formal approach to model hierarchical navigational structures of web pages. However, the semantic extensions of OOHDM in [10] is focused on the instantiation of a meta-model whereas the aggregation approach of [18] and the statechart approaches of [19, 9] only deal with the expansion of navigation models.

Addition, partitioning and positioning at the presentation layer are not directly addressed within traditional web modelling languages. Although OOHDM ([16]) uses Abstract Data Views and Abstract Data Objects to specify the arrangement of content on the screen, these concepts do not provide mechanisms for addition or partitioning. The HTML template language of Strudel ([7]) defines HTML tags which may be embedded into arbitrary HTML pages. These templates are rendered into HTML pages by the HTML generator of the Strudel system. As in OOHDM, Strudel also does not allow hierarchical decomposition of regions. Other approaches, like [5] in the field of adaptive hypermedia, allow the conditional activation and deactivation of content and links. However, these systems usually adapt existing HTML pages and are not focused on the modelling of presentation aspects. As a result, they do not support addition, partitioning and positioning at the design level.

As a result, hierarchical modelling has not been thoroughly addressed until now. Although the conceptual modelling of object oriented systems specify extension and refinement criteria applicable at the content layer of web applications, hierarchical navigation and presentation models have not been extensively investigated. The presented approach is unique in that it provides support for the hierarchical modelling of content, navigation and presentation aspects by utilizing consistency criteria.

5 Conclusion

In this paper we have introduced the basic ideas of an approach to coordinate the design of related similar web sites. We have identified consistency criteria between a model description of a web site and its specialization through extension, refinement and re-structuring at the content, navigation and presentation layer of web applications. We have informally explained these criteria through a running example. Current and future work concentrates on (1) expanding these basic ideas within the framework of a comprehensive web modelling language, (2) developing a corresponding formal model and associated formal consistency criteria and (3) the development of a set of tools for the design of web site families that support the designer in meeting the consistency cri-

teria.

References

[1] Aiguier, M. "ÉTOILE-specifications: An Object-oriented Algebraic Formalism with Refinement." *J. Log. Comput.*, V14, N2, pp. 145-178, 2004.

[2] Booch, G., Rumbaugh, J.E., Jacobson, I., "The Unified Modeling Language User Guide", *J. Database Manag.*, V10, N4, pp. 51-52, 1999.

[3] Ceri, S., Fraternali, P., Bongio, A., "Web Modeling Language (WebML): A modeling language for designing Web sites" *Computer Networks*, V33, N1-6, pp. 137-157, 2000.

[4] Costal, D., Olivé, A., Teniente, E., "Relationship Type Refinement in Conceptual Models with Multiple Classification", *ER '01*, Yokohama, Japan, pp. 397-411, 2001.

[5] De Bra, P., Aerts, A., Berden, B., de Lange, B., Rousseau, B., Santic, T., Smits, D., Stash, N., "AHA! The adaptive hypermedia architecture", *HYPERTEXT '03*, Nottingham, UK, pp. 81-84, 2003.

[6] Hoschka, P., "Synchronized Multimedia Integration Language", *World-Wide Web Consortium Recommendation*, http://www.w3.org/TR/REC-smil/, June 1998.

[7] Fernandez, M., Florescu, D., Kang, J., Levy, A., Suciu, D., "Catching the boat with Strudel: experiences with a Web-site management system", *SIGMOD '98*, NY, USA, pp. 414-425, 1998.

[8] Ginige, A., Murugesan, S., "Guest Editors' Introduction: The Essence of Web Engineering-Managing the Diversity and Complexity of Web Application Development", *IEEE MultiMedia*, V8, N2, pp. 22-25, 2001.

[9] Leung, K.R.P.H., Hui, L.C.K., Yiu, S., Tang, R.W.M., "Modeling Web Navigation by Statechart", *COMPSAC '00*, Taipei, Taiwan, pp. 41-47, 2000.

[10] Lima, F., Schwabe, D., "Modeling Applications for the Semantic Web", *ICWE '03*, Oviedo, Spain, pp. 417-426, 2003.

[11] Merialdo, P., Atzeni, P., Mecca, G., "Design and development of data-intensive web sites: The Araneus approach", ACM *Trans. Internet Techn.*, V3, N1, pp. 49-92, 2003.

[12] Pastor, O., Abrahao, S.M., Fons, J., "An Object-Oriented Approach to Automate Web Applications Development", *EC-Web '01*, Munich, Germany, pp. 16-28, 2001.

[13] Rossi, G., Schwabe, D., Lyardet, F., "Web Application Models Are More Than Conceptual Models", *ER '99 (Workshops)*, Paris, France, pp. 239-253, 1999.

[14] Schrefl, M., Stumptner, M., "Behavior Consistent Refinement of Object Life Cycles", *ER '97*, Los Angeles, USA, pp. 155-168, 1997.

[15] Schwabe, D., Rossi, G., "The Object-Oriented Hypermedia Design Model", *Commun. ACM*, V38, N8, pp. 45-46, 1995.

[16] Schwabe, D., Rossi, G., "An Object Oriented Approach to Web-Based Applications Design", *Theory and Practice of Object Systems*, V4, N4, pp. 207-225, 1998.

[17] Troyer, O.D., Leune, C.J., "WSDM: A User Centered Design Method for Web Sites", *Computer Networks*, V30, N1-7, pp. 85-94, 1998.

[18] Valderas, P., Fons, J., Pelechano, V., "Extending Navigation Modelling to Support Content Aggregation in Web Sites", *ICWE '04*, Munich, Germany, pp. 98-102, 2004.

[19] Winckler, M., Palanque, P.A., "StateWebCharts: A Formal Description Technique Dedicated to Navigation Modelling of Web Applications", *DSV-IS '03*, Funchal, Portugal, pp. 61-76, 2003.