Adaptation and Reuse in Designing Web Information Systems

Flavius Frasincar, Peter Barna, Geert-Jan Houben
Eindhoven University of Technology
PO Box 513
NL-5600 MB Eindhoven, the Netherlands
{flaviusf, pbarna, houben}@win.tue.nl

Zoltán Fiala
Dresden University of Technology
Mommsenstr. 13
D-01062 Dresden, Germany
zoltan.fiala@inf.tu-dresden.de

Abstract

The increasing number of requirements for a Web Information System asks for an engineered process in designing such a system. In this paper we focus on two of these requirements: presentation adaptation based on user preferences/device capabilities and reusability of the different design artifacts. Hera is a model-based design methodology for Web Information Systems. Adaptation and reuse can be tackled at different design levels in Hera. We illustrate by means of examples how adaptation and reuse can be achieved in the conceptual model, the application model, and the presentation model. Based on a static user profile, adaptation is realized by attaching appearance conditions to model elements. With respect to reusability we do focus on one mechanism that supports it, namely inheritance.

1. Introduction and related work

In order to cope with the increasing number of requirements for a Web Information System (WIS) several methodologies have been proposed for its design. Based on good engineering practices, the consideration of these requirements at an early step of the design process will ensure the application's success. Among these requirements we mention the ability of a WIS to support: presentation adaptation (e.g. based on browsing platform or user's preferences), reusability, maintenability, evolution etc. Building WISs that satisfy these requirements is far from being trivial.

A lot of effort is required for the maintenance and evolution of WISs. This is mainly due to the lack of abstraction primitives to capture the system design. Model-based design methodologies provide high level abstractions for representing content, navigation, and layout. The advantages of a model-based design are countless: better communication between the different stakeholders, the possibility to reverse engineer a WIS, the ability to reuse different parts of an application in a project but also between projects etc.

In previous work we have proposed Hera [6, 12], a model-based WIS design methodology using Semantic Web technology. It has all the benefits of a model-based methodology as well as the good traits of a Semantic Web product (e.g. platform independence, explicit semantics, interoperability etc.). The focus of this paper is on static adaptation (i.e. adaptation that does not consider the user's browsing behaviour) and on reusability as integrant parts of the Hera design process. Both adaptation and reuse have a great impact on WIS requirements such as maintenance and evolution and therefore contribute to the quality of the overall design process.

While there are powerful tools like AMACONT [3] that consider adaptation at implementation level, there is a lack of good WIS design specifications to include adaptation aspects. Compared with OOHDM [11] our proposed personalization goes one step further in the sense that we distinguish for each WIS design phase its "adaptation hot-spots", i.e. what adaptation aspects are relevant to this particular phase. The adaptation is achieved by attaching conditions [4] to the different design artifacts which resembles the Event-Condition-Action rules from WebML [2]. Differently than WebML we make the semantics of the different models explicit (i.e. part of the model) using RDF(S) [1, 9] as a modeling language, instead of XML.

As previously emphasized in [7] inheritance is a good mechanism to support the reuse of the different design artifacts. Inheritance can also support adaptation by appropriately subclassing design artifacts to satisfy different adaptation needs. RDF(S) has a built-in subclassing mechanism that enables to factor out shared resource properties into a common resource. This common resource can be reused in defining new resources decreasing thus the design effort required for producing these new resources. In RDF(S) reuse is also facilitated by the possibility of adding new properties to existing resources, multiply classifying resources, refining properties etc. It is due to its flexibility and extensibility capabilities that we chose to represent our models in RDF(S).

2. Hera Methodology

Hera [6, 12] is a model-based methodology for designing WISs. Based on the principle of separation of concerns it distinguishes three design models: the conceptual model, the application model, and the presentation model. Each model captures a different aspect of the system: the conceptual model specifies the data (content) that needs to be presented, the application model captures the application logics (navigation), and the presentation model deals with layout related issues. A characteristic feature of the considered WISs is the fact that data is not known in advance (the data can be obtained, for example, as a result of a user query). Nevertheless, we assume that the data's schema is known and as a consequence all Hera models will be at schema level.

Some of the models are overlay models of the previous ones in the Hera methodology steps. The application model extends the conceptual model with navigation primitives and the presentation model enhances the application model with layout primitives. It is exactly this superposing of models that enables the pipeline transformations between model instances [5]. In Hera we use a graphical notation to represent the different models that has a natural RDF/XML serialization. In the rest of this paper we will present the Hera models mainly in graphical notation as it easier for the reader to grasp specifications in a graphical way than in the verbose RDF/XML serialization.

2.1. Conceptual model

The conceptual model (CM) is the schema of the data that needs to be presented. CM is composed of concepts, attributes, and relationships. Attributes relate concepts to media types and relationships relate concepts to each other. Each relationship has its cardinality specified as well as its inverse relationship. The running example used through the rest of the paper is based on a art museum. Figure 1 depicts the CM for our example.

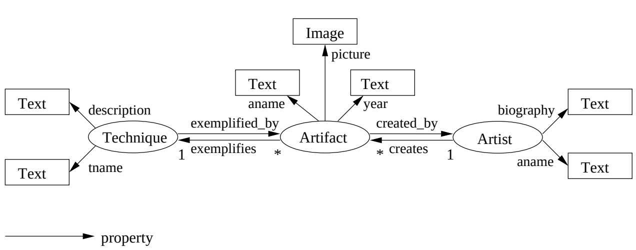

Figure 1. Conceptual model

Part of the CM is the media model (MM). In the same way as AMACONT [3] we base our MM on a subset of the MPEG-7 standard [10]. In Figure 2 the considered media types: text, image, audio, and video have corresponding RDF representations that capture their relevant properties. Note that media items are referred via a URL and all their properties are pointing to literals. Adding a new media type will be seamlessly done by subclassing the appropriate (parent) media type and by adding its characteristic properties.

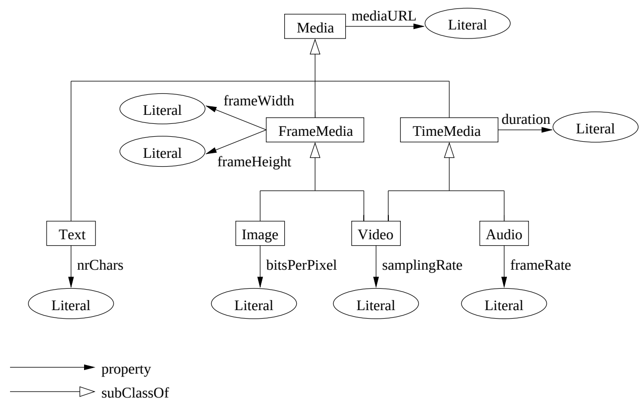

Figure 2. Media model

2.2. Application model

The application model (AM) represents a special view over the data schema. It is not a classical database view in the sense that it adds navigation primitives to the model. AM is composed of slices and slice relationships. A slice is a meaningful data presentation unit. Slices can be aggregated by means of compositional relationships, and the navigation between slices is defined by navigational relationships. Each slice is owned by a concept or in other words a slice is a new concept property. The most primitive slices are the slice attributes which correspond to the concept attributes from CM. Figure 3 illustrates a part of the AM for the museum example.

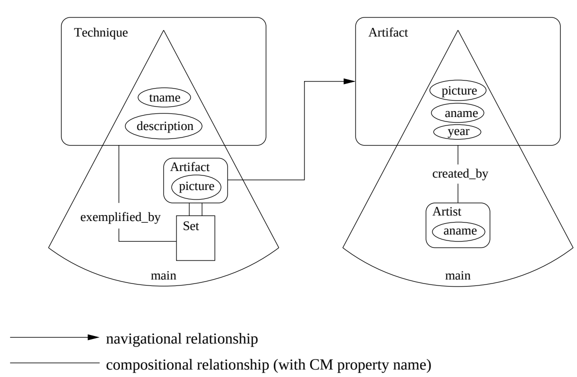

Figure 3. Application model

2.3. Presentation model

The presentation model (PM) describes the presentation's layout. PM is composed of regions that represent a rectangular area on the user's display. While slices are associated to concepts, regions are associated to slices. Navigational relationships from AM are materialized to different region relationships: spatial relationships (compositional relationships from AM are good candidates for spatial relationships), temporal relationships, and hyperlinks (navigational relationships from AM are good candidates for hyperlinks). Spatial relationships can be specified quantitatively by giving the (x,y) coordinate with respect to the top-left corner of the parent region or qualitatively by specifying a qualitative constraint (e.g. "right") in relation to another region. Figure 4 presents one region from the PM in our running example. In order not to complicate the figure we omit from it the region relationships between attributes.

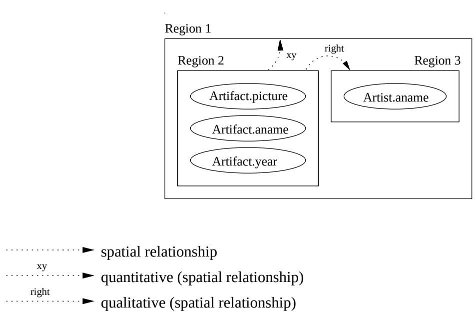

Figure 4. Presentation model

3. Adaptation

The presentation adaptation considers "adaptation hot-spots" [11] during a WIS design. Basically all Hera models considered in section 2 are adaptation hot-spots. In this paper we consider only static adaptation, i.e. an adaptation that does not consider the user's browsing behaviour.

The static adaptation is based on another model, the user profile, a CC/PP [8] vocabulary to model user preferences and device capabilities. An excerpt of a user profile instance is given below

```
<Description rdf:about="Profile">
 <ccpp:component>
  <HardwarePlatform>
   <imageCapable>Yes</imageCapable>
   <client>Desktop</client>
   ...
  </HardwarePlatform>
 </ccpp:component>
 <ccpp:component>
  <UserPreferences>
   <levelOfExpertise>Expert</levelOfExpertise>
   ...
  </UserPreferences>
  </ccpp:component>
    ...
</Description>
```

The adaptation is realized by means of appearance conditions attached to different design artifacts. Evaluating such conditions to true/false enables/inhibits the presence of their associated artifacts in the design. Because of the overlay nature of Hera models, an artifact deleted in one model will also be eliminated from the subsequent models. In order to specify meaningful adaptation conditions, each model can only use a subset of the profile attributes in conditions.

3.1. Adaptation in CM

Adaptation in CM removes concepts and media types that have an associated condition not valid. Figure 5 depicts an adaptation condition in CM (remember that MM is part of CM). The media items corresponding to the `Image` media type will be part of the CM instance only if the user's device has image viewing capabilities. The same adaptation technique can be used on the concepts from Figure 1.

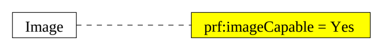

Figure 5. Adaptation condition in CM

3.2. Adaptation in AM

Adaptation in AM suppresses slices that do not fulfill an attached condition. As a consequence navigation relationships that are pointing to suppressed slices will be hidden. Figure 6 presents an adaption condition based on the level of expertise of a user (`Beginner`, `Normal`, or `Expert`) with respect to a given domain. If for example the user is an `Expert` he will have access to the textual description of a certain artistic technique in slice `Technique.main`.

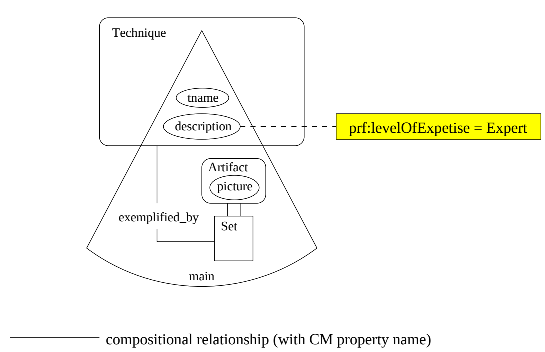

Figure 6. Adaptation condition in AM

3.3. Adaptation in PM

Adaptation in PM eliminates regions that have an assigned condition invalid. As a side effect region relationships that involved the removed regions will also be discarded. Figure 7 shows two mutual exclusive conditions depending on what kind of client (`Desktop` or `WAP Phone`) the user has. For a `Desktop` client the available horizontal space is larger than for example a `WAP Phone` client. As a consequence the region displaying the artist's name is on the right hand side of the artifact's region for a `Desktop` client and below the artifact's region for a `WAP Phone` client.

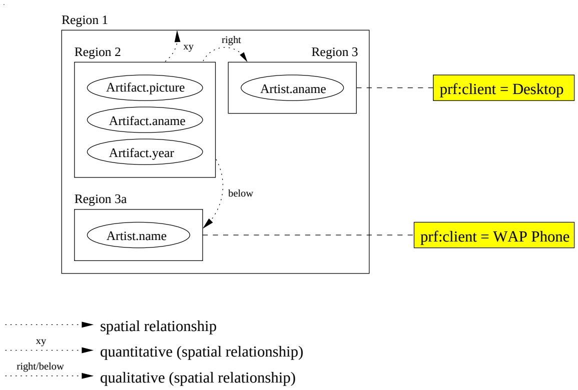

Figure 7. Adaptation condition in PM

4. Reuse

All Hera models are suitable for reuse by means of different recycling mechanisms. There are a lot of recycling mechanisms among which we mention inheritance [7], prototyping (code sharing) [3, 7], attaching new properties to existing design artifacts etc. In this paper we focus on inheritance. Extending a model by means of inheritance will enable also the (direct) reuse of design aspects of the old subsequent models if the designer didn't refine them (indirect reuse).

4.1. Reuse in CM

Concepts from CM can be easily extended to new concepts by subclassing the old ones. Figure 8 illustrates the insertion of two new concepts `Painting` and `Painter` as extensions of the existing `Artifact` and `Artist` concepts. The `Painting` concept has the new `area` property attached to characterize the painting's surface. The concept relationships `creates` and `created_by` are also appropriately refined by `paints` and `painted_by`. In subsection 2.1 we already discussed how a new media type similar to an old one can be added to MM based on inheritance.

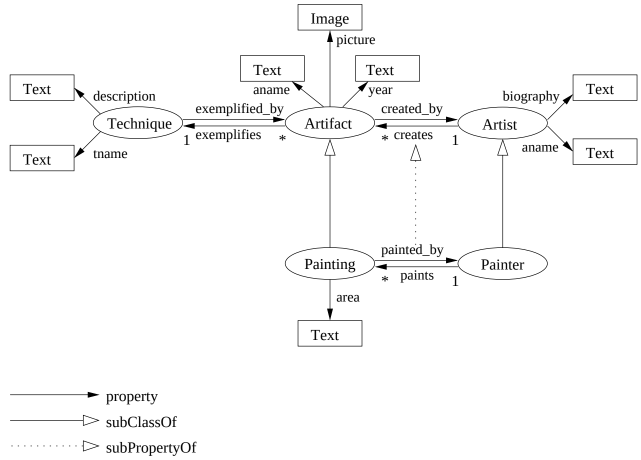

Figure 8. Inheritance in CM

4.2. Reuse in AM

Slices from AM can be extended to new slices that increase the complexity of the old ones by adding new slice compositional elements and new navigational relationships. For example, these new slices can add to the AM the properties of the new concepts inserted to the CM in subsection 4.1. Figure 9 depicts the subclassing of the `Artifact.main` slice by the `Painting.emain` slice.

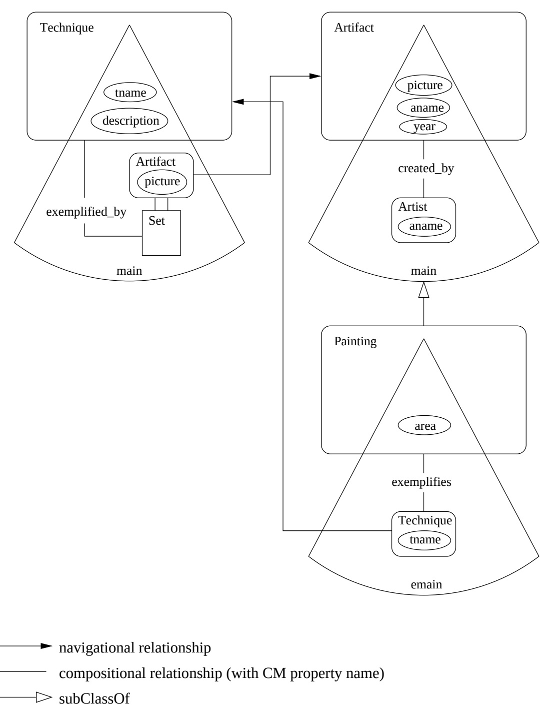

Figure 9. Inheritance in AM

Painting.emain contains in addition to the old compositional elements inherited from its parent slice, the painting's area and the painting's technique tname. Moreover, Painting.emain offers the possibility to navigate from the painting's technique tname to the Technique.main slice.

4.3. Reuse in PM

Regions from PM can be refined by adding new regions and new region relationships. The resulting regions can handle the increased AM complexity from subsection 4.2 and they can also be used to produce more sophisticated layouts. Figure 10 exemplifies the refinement of two regions. Region 1r refines Region 1 by adding Region 2r and Region 4 to its definition. Region 2r is a subclass of Region 2 and displays painting specific attributes. The newly inserted Region 4 has a below spatial relationship with the old Region 3.

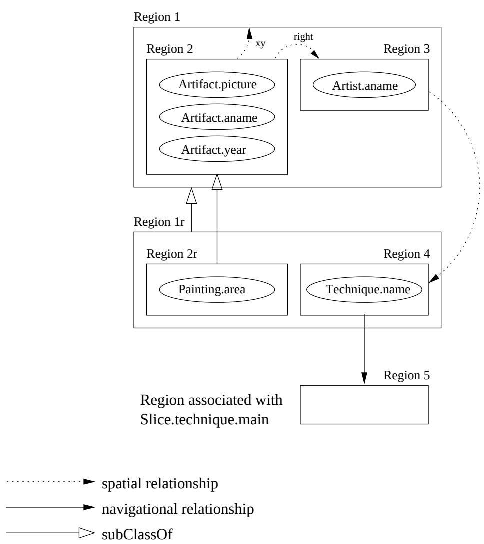

Figure 10. Inheritance in PM

5. Conclusions

In this paper we have presented how adaptation and reuse can be supported by the Hera WIS design methodology. Based on a user profile we showed using examples how conditions attached to different design artifacts achieve the removal of different media types from CM, the exclusion of slices and slice relationships from AM, and the elimination of regions and region relationships from PM. Also by means of examples we emphasized how inheritance fosters reuse of previously designed CM concepts, AM slices, and PM regions.

We strongly believe that a methodology that supports application requirements as early as possible in its design phase will ensure its success. As future work we would like to extend the Hera methodology to address (at design level) other WIS requirements like adaptivity (which considers user's browsing behaviour) or user interaction (which takes in account user's input to the system).

Furthermore we would like to complement Hera's specification models with the flexible AMACONT implementation. The adaptation aspects specified in this paper correspond naturally to the variant selection mechanism of AMACONT components.

References

[1] D. Brickley and R. V. Guha. Rdf vocabulary description language 1.0: Rdf schema. W3C Working Draft 10 October 2003.

[2] S. Ceri, P. Fraternali, and A. Bongio. Web modeling language (webml): a modeling language for designing web sites. *Computer Networks, Ninth International World Wide Web Conference*, 33:137–157, 2000.

[3] Z. Fiala, M. Hinz, K. Meissner, and F. Wehner. A component-based approach for adaptive, dynamic web documents. *Journal of Web Engineering*, 2(1-2):58–73, 2003.

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

[5] F. Frasincar, G. J. Houben, P. Barna, and C. Pau. Rdf/xml-based automatic generation of adaptable hypermedia presentations. In *International Conference on Information Technology: Coding and Computing, Track on Modern Web and Hypermedia Systems*, pages 410–414. IEEE Computer Society, 2003.

[6] F. Frasincar, G. J. Houben, and R. Vdovjak. Specification framework for engineering adaptive web applications. In *The Eleventh International World Wide Web Conference, WWW 2002, Web Engineering Track*, 2002.

[7] H.-W. Gellersen and M. Gaedke. Object-oriented web application development. *IEEE Internet Computing*, 3(1):60–68, 1999.

[8] G. Klyne, F. Reynolds, C. Woodrow, O. Hidetaka, J. Hjelm, M. H. Butler, and L. Tran. Composite capability/preference profiles (cc/pp): Structure and vocabularies. W3C Working Draft 25 March 2003.

[9] O. Lassila and R. R. Swick. Resource description framework (rdf) model and syntax specification. W3C Recommendation 22 February 1999.

[10] J. M. Martinez. Mpeg-7 overview. Version 9, ISO/IEC JTC1/SC29/WG11/N5525 March 2003.

[11] D. Schwabe, G. Rossi, and S. D. J. Barbosa. Designing personalized web applications. In *Tenth International World Wide Web Conference*, pages 275–284. ACM, 2001.

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