Design and Development of Component-based Adaptive Web Applications

Kurzfassung der Dissertation

zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

vorgelegt an der
Technischen Universität Dresden
Fakultät Informatik

eingereicht von

M.Sc. Zoltán Fiala
geboren am 19. März 1977 in Budapest

Betreuender Hochschullehrer: Prof. Dr.-Ing. Klaus Meißner
Technische Universität Dresden

Dresden, den 16. Oktober 2006

1 Introduction and Objectives

Recently, the amount of information published on the WWW has grown enormously. The survey in [Lyman et al. 2003] reports of 167 Terabytes of Web content and prognosticates an annual doubling for the future. Another trend is the Web's evolution to a dynamic interaction and application medium. Formerly a collection of static HTML pages, today's Web sites are complex *Web Information Systems (WIS)* that offer up-to-date content and functionality. Their constantly growing user community is characterized by heterogeneous goals, preferences, and capabilities. Furthermore, people access the Web from a diversity of locations and (mobile) client devices, such as cell phones, PDAs, or set-top boxes. These trends necessitate *adaptive Web sites* that automatically adjust themselves to the specifics of their audience.

While the development of early Web systems was characterized by the ad hoc authoring of Web pages, the aforementioned trends have soon made clear that this straightforward approach is not sufficient for complex Web sites. The recently emerged *Web engineering* field tackles this problem by the establishment of "sound scientific, engineering and management principles" to Web site development [Murugesan and Deshpande 2001]. In the last decade, several Web design methods (e.g. OOHDM [Schwabe et al. 1996], WebML [Ceri et al. 2000], WSDM [De Troyer 2001], Hera [Vdovjak et al. 2003]) have been proposed by academia. While using different techniques and notations, they are common in distinguish between a data, a navigation, and a presentation design. Selected methods also offer some support for adaptation at design-level, yet mostly centered around certain content and navigation adaptation aspects. Furthermore, there is only limited support for the visual specification of adaptation and the (semi-)automatic generation of a corresponding context-adaptive implementation.

Another shortcoming is the current implementation model of the WWW. While well applicable for easy Web page authoring, it is obviously not a sufficient implementation base for structured Web engineering [Gaedke et al. 2000]. Even though some Web design methods provide a (semi-)automatic implementation generation, fine-grained Web design artefacts get lost during implementation when being transformed to conventional Web document formats such as (X)HTML, cHTML, WML documents, etc. These formats do not facilitate a clear separation of concerns, like content, semantics, navigation, and presentation. Furthermore, they provide no support to describe the adaptive behavior of reusable content pieces in a generic way. This prevents the reuse of independent and adaptable implementation artefacts both within an application and for other applications or target systems. As a consequence, most providers create and manage Web code for different platforms and contexts separately.

At the same time, the efficient reuse of configurable code is a main task of traditional software technology and has been successfully addressed by component-based software engineering [Szyperski 1998]. Component-based approaches have numerous advantages: reusability, system-independence, configurability, composability, etc. Traditional component models consider components as binary units of composition. Still, in recent years, different approaches have been proposed to apply their principles to the document-centric nature of Web and multimedia applications [Gaedke et al. 2000, Wehner and Lorz 2001, Dachselt et al. 2002]. While all these approaches benefit from the reuse of declarative, configurable implementation artefacts in a component-like manner, none of them supports the adaptation and personalization issues mentioned above. Moreover, there is a lack of solutions for the automatic generation of a component-based implementation based on high-level Web design specifications.

Objectives: Inspired by a thorough review of related Web engineering approaches, this dissertation addresses the shortcomings mentioned above by combining the benefits of model-based Web design methods with the advantages of component-based implementation techniques for efficiently engineering *adaptive* Web sites. The vision is the intuitive composition of context-dependent Web applications from declarative, reusable, and adaptable components, aided by a systematic development process and appropriate tool support. To fulfill this vision, a novel, *concern-oriented component model* is proposed for adaptive Web applications. Furthermore, a *multi-stage, model-based authoring process* and a visual authoring tool for the efficient development of adaptive Web components are introduced and constructively validated by a number of example applications. Finally, it is investigated how the lessons learned from engineering component-based Web sites can be generalized to add adaptation to a broader range of existing (not component-based) Web applications.

2 A Concern-oriented Component Model for Adaptive Web Applications

To provide component-based reuse in adaptation engineering, this thesis proposes a concern-oriented component model for adaptive dynamic Web applications [Fiala et al. 2003]. The term *concern-oriented* denotes its explicit support for a clear separation of concerns involved in a Web application (e.g. content, navigation, semantics, presentation, as well as their adaptation aspects), each being dealt with on different levels of abstraction[1]. The component model is fully declarative and relies on existing Internet standards, e.g. XML Schema, MPEG-7, XPath, and XPointer. Its layered architecture is depicted in Figure 1.

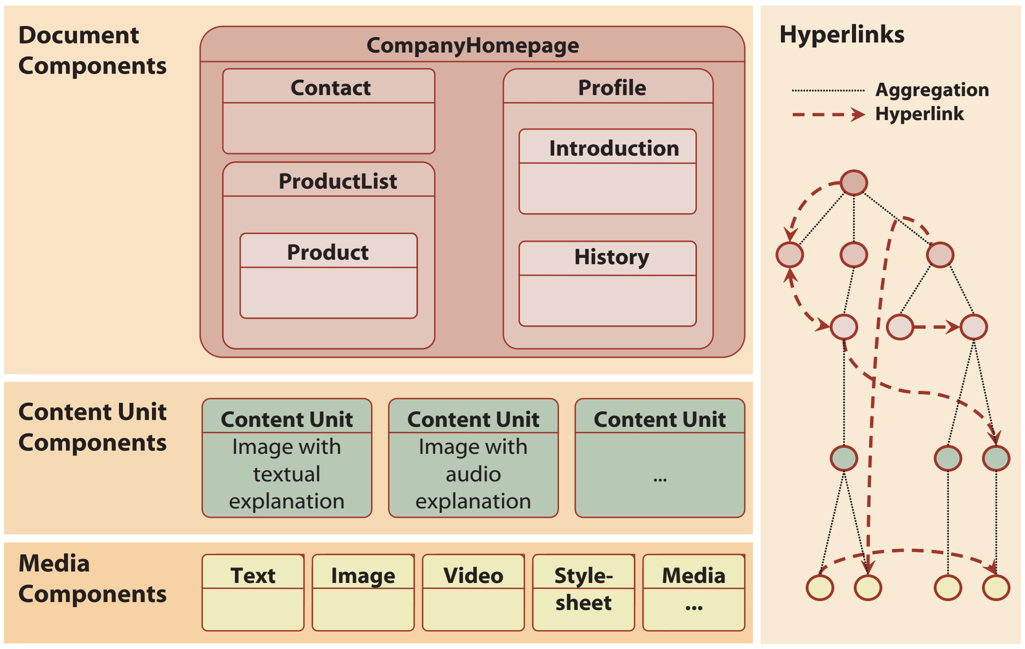

Figure 1: A concern-oriented component model for adaptive Web sites [Fiala et al. 2003]

[1]The component model was developed within the scope of the AMACONT project and is thus also referred to as the AMACONT component model. The thesis presents the author's contributions to it based on requirements for the efficient authoring of adaptive Web applications from reusable components.

On the lowest level, there are *media components* encapsulating concrete media assets, such as text, structured text (e.g. HTML), images, videos, Java applets, etc. The second level combines media components that belong together semantically - e.g. an image with a textual description - to so-called *content unit components*. Defining such collections of media components is a key reuse factor. Third, *document components* are specified as parts of Web applications playing a well-defined semantic role (e.g. a news column or a product presentation). They can either reference content units or aggregate other document components. Finally, the orthogonal *hyperlink view* defines typed links spanned over all component levels.

The component model provides different adaptation mechanisms. First, it facilitates the *self-adaptation* of components by the definition of adaptation variants for all important component aspects (e.g. content, structure, navigation) on various abstraction levels. The selection of a given alternative is determined based on inherent adaptation rules contained in components, i.e. they automatically adjust themselves to the current usage context. Second, the *adaptive layout* of components is specified by abstract layout descriptors (e.g. *BoxLayout*, *BorderLayout*, *OverlayLayout*, or *GridLayout*) that can be automatically adapted to different output formats. They describe a size- and client-independent layout, allowing to abstract from the exact resolution of a display or browser window. Third, there is support for so-called *component templates* that can be expanded by context-dependent queries to dynamic data sources. This latter mechanism allows for the composition of data-driven adaptive Web information systems. By a number of examples, it is shown how these basic adaptation facilities can be combined to implement the most important content, navigation, and presentation adaptation techniques identified by the adaptive hypermedia community [Brusilovsky 2001].

At run-time, components are dynamically transformed to hypermedia presentations, adapted to the user, his client, and entire usage context. The clean separation of concerns allows to utilize a stepwise transformation process, each stage being responsible for a given application or adaptation aspect. The resulting document generation pipeline supports different platforms and Web output formats, such as (X)HTML, cHTML, and WML [Hinz et al. 2004].

3 The Multi-Stage Development Process and its Tool Support

The concern-oriented component model provides a flexible implementation layer for adaptive Web applications. Still, while component-based reuse is crucial to Web engineering, the development of adaptive Web sites out of such components is a complex task requiring structured design processes and appropriate tool support. The thesis addresses this issue by combining the benefits of systematic, model-based design methods with the reuse and adaptation capabilities of concern-oriented components. The corresponding scientific contributions comprise 1) the design of a structured authoring process for component developers, 2) the development of a graphical authoring tool, and 3) the design and prototypical implementation of a transformation architecture for the model-driven generation of component-based Web applications.

3.1 Hera-AMACONT: Structured Component Authoring based on a Hypermedia Design Method

Independent of a given application domain, the concern-oriented component model supports different Web application types (from static adaptive hypermedia presentations to dynamic Web information systems). Thus, instead of suggesting an own methodology, the chosen strategy is to apply existing design methods for the structured development of component-based adaptive Web applications. The thesis focuses on an important development scenario: the engineering of data-driven adaptive Web information systems from reusable components. Therefore, it adopts and extends the model-based Hera design method [Vdovjak et al. 2003] to the context of component-based Web engineering. Considering the design steps identified by the Hera models, it is shown how component authors can implement them by creating, configuring, aggregating, and interlinking document components.

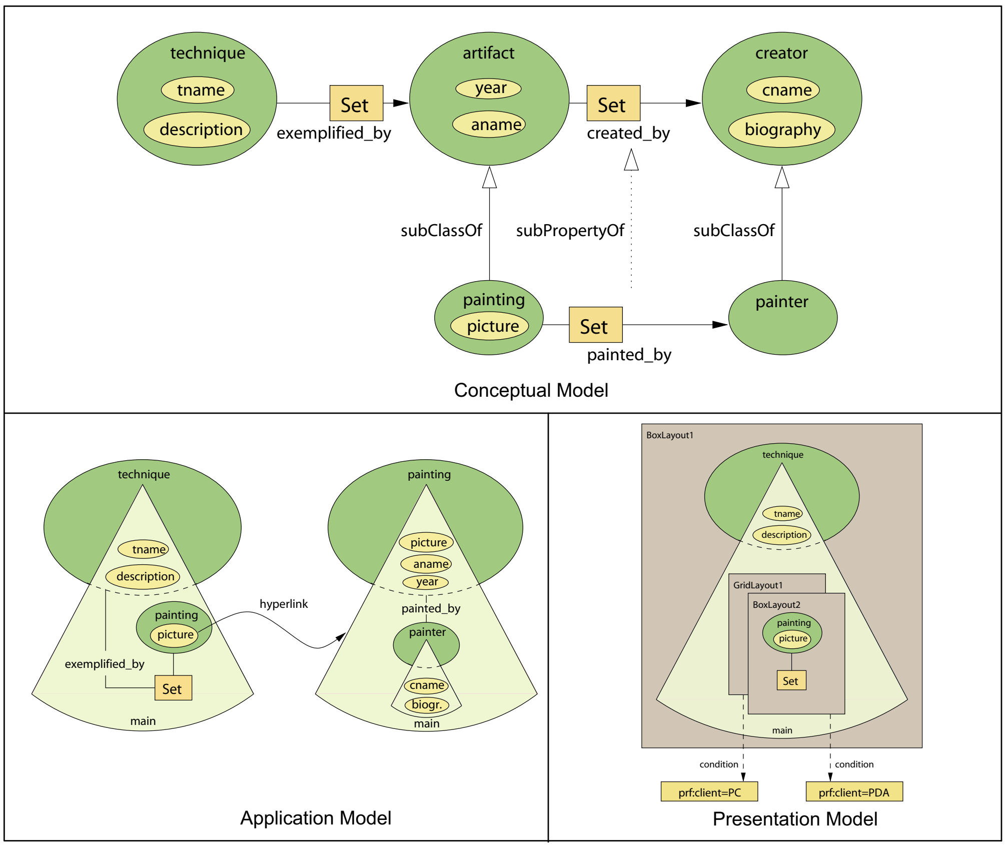

Figure 2: The design phases of the Hera-AMACONT methodology

The resulting engineering process is called Hera-AMACONT and distinguishes three design phases. Using semantic Web modeling techniques, each addresses a certain application concern in an implementation independent way (see Figure 2). The *conceptual design* phase models the application domain by a hierarchy of concepts, their attributes, and relationships, thus tackling both semantic and content adaptation. The *application design* phase defines the hypermedia structure of the planned Web application by specifying abstract navigational units (so-called slices), their relationships (aggregation and interlinking), as well as structural and navigational adaptation. Finally, the *presentation design* phase specifies the interface aspects of a Web application by the implementation-independent Presentation Model (PM). It defines the (adaptive) spatial arrangement of the formerly identified navigational units based on rectangular regions and their corporate design in terms of style elements.

Hera-AMACONT extends the original Hera models by the design of a presentation model addressing different kinds of static and dynamic adaptation [Fiala et al. 2004b]. It explicitly tackles context-dependency at presentation modeling, a concern that has not been sufficiently addressed by previous methodologies. Thus, it systematically deals with separate application and adaptation aspects (regarding data, navigation, presentation) at design time.

Furthermore, for each design phase a corresponding "component-based implementation recipe" is identified [Fiala et al. 2004a]. Considering those design steps (models) as a "guideline", it is shown how component authors can map them to a component-based Web application by systematically creating, configuring, aggregating, and interlinking reusable document components. Thus, a (possible) structured model-based authoring process is designed for the developers of component-based adaptive Web information systems.

3.2 A Modular Authoring Tool for Component-based Adaptive Web Sites

To put a given design or authoring process (such as the one dictated by Hera-AMACONT) into practice, component authors need appropriate tool support for creating, configuring, and aggregating components. Therefore, a graphical component authoring tool called the AMACONTBuilder was developed [Fiala et al. 2005]. Again, a basic requirement was the tool's flexibility to cope with different application scenarios and possible process models. To this end, a modular architecture was established, allowing to assign an extensible set of graphical editor plug-ins to different component types and properties (see Figure 3).

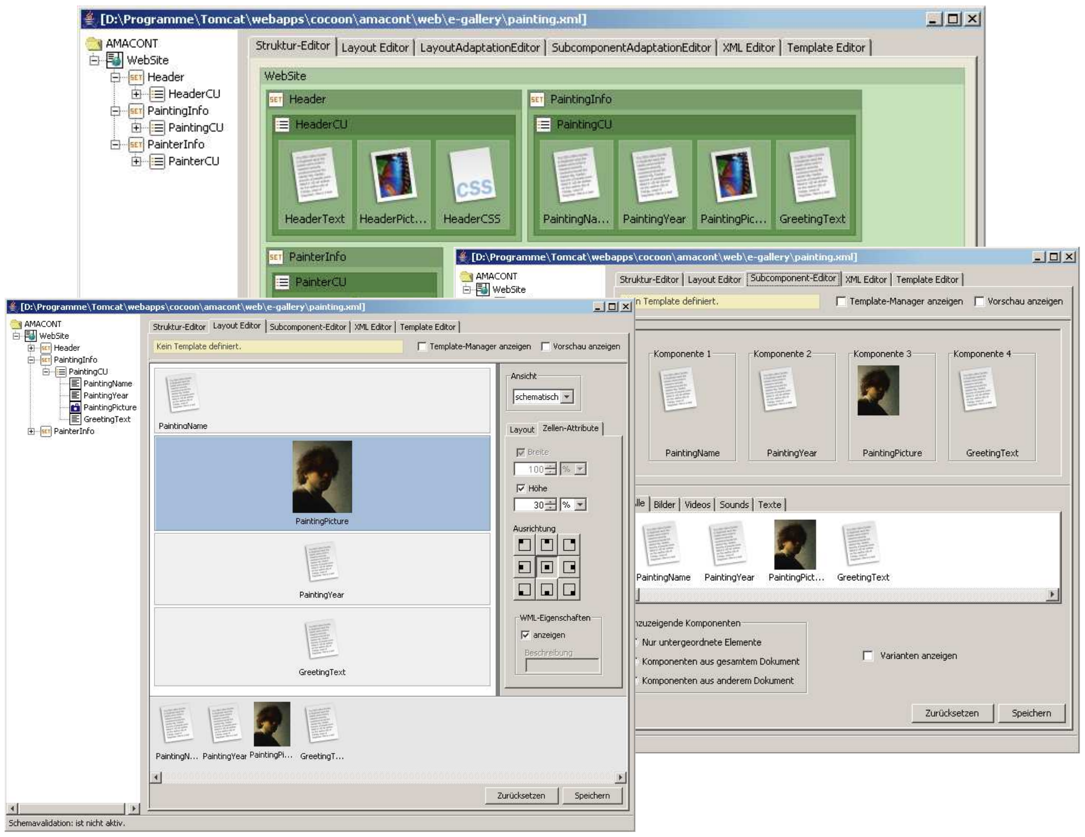

Figure 3: The authoring tool AMACONTBuilder [Fiala et al. 2005]

The AMACONTBuilder's *media component editors* support the visual creation and configuration of media components. By means of its *editors for hypertext authoring*, media components can be aggregated to composite components and interlinked to complex hypermedia structures. Similarly, the *layout editor* and the *style editor* allow to configure the visual appearance of component structures. All editor modules are provided with a generic facility to visually define adaptation variants and corresponding rules based on the current usage context. Furthermore, flexible authoring workflows are supported for different application types, authoring roles, and process models (i.e. also authoring processes different from Hera-AMACONT). The thesis introduces the tool's main functionality from a developer's point of view, providing details on its technical realization.

3.3 From Component Authoring to Model-driven Implementation Generation

As an implementation-centric tool being independent of a specific design method or authoring process, the AMACONTBuilder facilitates flexible component authoring. However, in some cases it is desirable to take a further step from model-based component authoring to model-driven component generation, thus adding automation to the overall process of design and component-based implementation. Therefore, the dissertation also deals with the research question of how high-level design specifications (models) can be automatically translated to a component-based adaptive Web presentation in a model-driven way.

After investigating the requirements for the automation of the formerly identified mappings between high-level design models and concern-oriented components, an RDF(S)-based formalization of the Hera-AMACONT presentation model is provided. Furthermore, a model-driven component generation architecture based on a series of transformation steps is designed and prototypically implemented. Taking as input a structured data source, it creates a component-based Web application in a stepwise manner, with each transformation step incorporating a certain design model (i.e. a given application and adaptation aspect) into the resulting implementation. The component structures generated this way manifest all application and adaptation design issues dictated by the underlying models.

3.4 Putting it all together

Figure 4 shows an overview of the resulting multi-stage Web engineering process. As depicted there, it starts on the highest abstraction level of *design and modeling*, resulting in a model-based specification of the targeted Web information system according to the Hera-AMACONT methodology. Such a design can be mapped to the second level of *component-based implementation* either automatically in a model-driven way or manually by the AMACONTBuilder. The flexibility of the AMACONTBuilder also allows to proceed according to another process design model or even to further "refine" an automatically generated component-based implementation. Finally, on the lowest level of *document generation*, the resulting component structures are automatically published to various Web output formats and adapted to the current usage context.

The overall authoring process and its tool support were constructively validated by a number of adaptive multimedia Web application prototypes: an adaptive painting gallery, an online music store prototype, as well as an adaptive Web information system providing information on students' works at the author's university. The dissertation explains the different phases and tools of designing and developing component-based adaptive Web applications based on these examples, respectively.

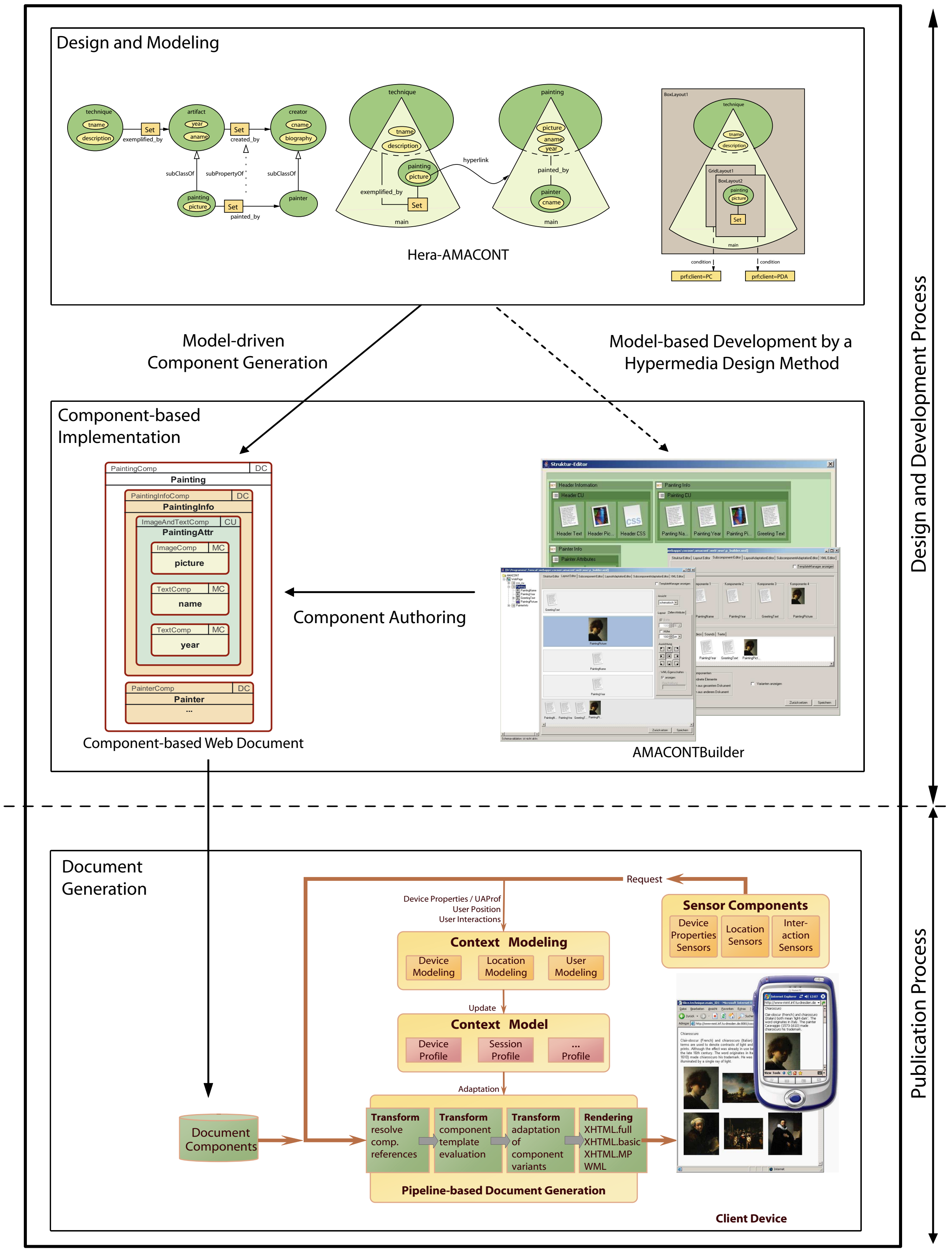

Figure 4: Overview of the multi-stage development process

4 Generalization with the Generic Adaptation Component

The combination of the concern-oriented component model with a structured, model-based design method provides an efficient means for developing adaptive Web-based systems. Still, the resulting engineering process assumes to build adaptive Web applications "from scratch", not supporting developers aimed at adding adaptation to existing applications. Therefore, the final part of the dissertation addresses the research question of how the previous results can be generalized for extending existing Web applications with additional adaptation concerns. The key observation is made that adaptive Web applications can be reduced to a series of data transformations and that both major parts of the adaptation specific transformations as well as their configuration can be separated from the rest of an application. Consequently, it is possible to realize selected adaptations based on *generic* transformers that are controlled by an *external* configuration. Thus, the concept of document components containing inherent adaptation descriptions can be generalized for a broader range of Web sites by the external assignment of adaptation rules to fragments of XML-based document formats.

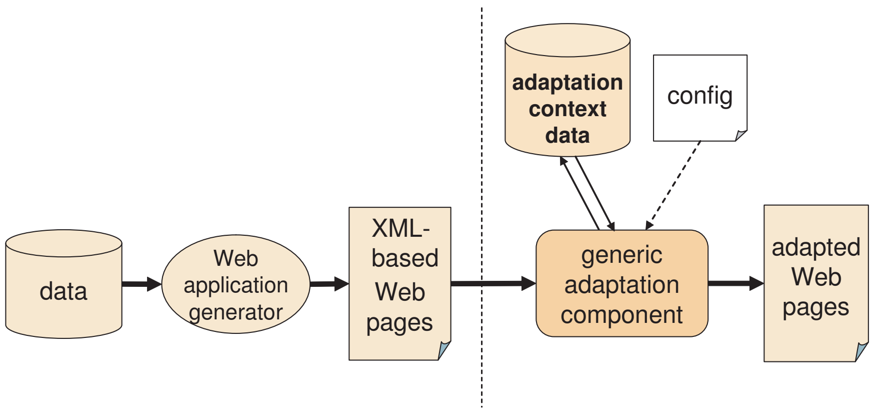

Figure 5: GAC abstract system overview

To prove this concept, a transcoding tool called the Generic Adaptation Component (GAC [Fiala and Houben 2005]) is introduced (Figure 5). It processes XML-based content provided by an arbitrary *Web application generator* and adjusts it to the characteristics of individual users and their clients. As a generic component, the GAC can perform different adaptations on its input, the recipe for which is specified by its RDF-based *configuration* language. This configuration is based on an extensible taxonomy of generic *adaptation rules* and *context data update rules*, each dictating and implementing a different content adaptation aspect. Those rules can reference arbitrary parameters from the *adaptation context data* to which the GAC has both read and write access. Based on a number of examples, it is illustrated how GACs can be configured to easily add different adaptation concerns to existing Web applications in an aspect-oriented manner [Fiala and Houben 2005].

5 Conclusion

This dissertation focused on the model- and component-based design and development of adaptive Web applications and introduced a novel, concern-oriented component-model for this purpose. It supports the definition of document components that encapsulate both separate application and adaptation concerns on different abstraction levels, i.e. its expressivity and reusability goes beyond the possibilities of conventional Web document formats. By a number of examples, it was shown that it is applicable for implementing the most important hypermedia adaptation techniques. Furthermore, it was illustrated how document components can be automatically transformed into different Web document formats, adapted to the appropriate user and his usage context.

The applicability of the proposed component model in the overall engineering process of adaptive Web applications was also successfully proved. Its combination with the Hera-AMACONT methodology provides significant research benefits: 1) the thorough separation of concerns and the reuse of artefacts at both design and implementation, 2) the systematic consideration of different adaptation aspects at each development step, and 3) the design-time support for presentation layer adaptation, an issue that has not been sufficiently addressed by existing methodologies, previously. With the AMACONTBuilder a flexible visual tool was introduced to facilitate different authoring scenarios. Furthermore, the RDF formalization of the Hera-AMACONT presentation model enabled even the model-driven generation of a component-based implementation. The concepts and tools of the overall multi-stage authoring process were demonstrated and thus constructively validated by a number of prototypes. To our best knowledge, the presented approach is the first development method for adaptive Web applications that combines the advantages of model-based Web design methodologies (high-level specification, clear separation of design concerns) with the benefits (e.g. reusability, configurablity, self-adaptation, etc.) of a component-based implementation techniques.

Finally, the research goal of extending existing XML-based Web applications with (additional) adaptation concerns was also achieved. Aided by the Generic Adaptation Component, Web developers can decouple selected adaptation operations from the rest of a Web application and specify them at a later stage, i.e. after the Web site has already been deployed. As this specification of adaptation is not intertwined with the regular Web application, it allows easy re-use of adaptation configurations for different sites. Again, the developed concepts could be validated by an implementation based on example applications.

The work presented in this dissertation provides different opportunities for future research. First, possible extensions of the component model can be mentioned, among them the provision of additional adaptation facilities, the integration of streaming multimedia content, and the development of transformation stylesheets to alternative output formats (e.g. MPEG-4, PDF, fat client applications). Second, the extension of the model-based authoring process towards Rich Internet Applications (RIA) and the further development of the AMACONT-Builder to an interdisciplinary, collaborative authoring environment is also a promising challenge. Third, the utilization of the GAC as a "portable personal adaptation assistant" on the client side is also an interesting direction for future work. Finally, a challenging research issue to be examined in more detail is the application of principles of aspect-oriented programming (AOP) for the design of additional adaptation concerns for existing Web applications.

References

[Brusilovsky 2001] Peter Brusilovsky. Adaptive Hypermedia. *User Modeling and User Adapted Interaction*, 11(1-2):87–110, 2001.

[Ceri et al. 2000] Stefano Ceri, Piero Fraternali, and Aldo Bongio. Web Modeling Language (WebML): a modeling language for designing Web sites. In *WWW9*, May 2000.

[Dachselt et al. 2002] Raimund Dachselt, Michael Hinz, and Klaus Meißner. CONTIGRA: an XML-based architecture for component-oriented 3D applications. In *7th International Conference on 3D Web Technology (Web3D 2002)*, pages 155–163, Februar 2002.

[De Troyer 2001] Olga De Troyer. Audience-driven Web Design. In *Information Modeling in the New Millennium*, pages 442–462. IDEA GroupPublishing, 2001.

[Fiala and Houben 2005] Zoltán Fiala and Geert-Jan Houben. A Generic Transcoding Tool for Making Web Applications Adaptive. In *The 17th Conference on Advanced Information Systems Engineering (CAiSE'05)*, pages 15–20. FEUP, June 2005.

[Fiala et al. 2003] Zoltán Fiala, Michael Hinz, Klaus Meißner, and Frank Wehner. A Component-based Approach for Adaptive Dynamic Web Documents. *Journal of Web Engineering, Rinton Press*, 2(1&2):058–073, September 2003.

[Fiala et al. 2004a] Zoltán Fiala, Flavius Frasincar, Michael Hinz, Geert-Jan Houben, Peter Barna, and Klaus Meissner. Engineering the Presentation Layer of Adaptable Web Information Systems. In *ICWE2004, Munich*, pages 459–472, 2004.

[Fiala et al. 2004b] Zoltán Fiala, Michael Hinz, Geert-Jan Houben, and Flavius Frasincar. Design and Implementation of Component-based Adaptive Web Presentations. In *19th Symposium on Applied Computing (SAC2004), Nicosia, Cyprus*. ACM Press, 2004.

[Fiala et al. 2005] Zoltán Fiala, Michael Hinz, and Klaus Meissner. Developing Component-based Adaptive Web Applications with the AMACONTBuilder. In *7th IEEE International Symposium on Web Site Evolution (WSE2005), Budapest, Hungary*, pages 39–45, 2005.

[Gaedke et al. 2000] Martin Gaedke, Christian Segor, and Hans-Werner Gellersen. WCML: Paving the Way for Reuse in Object-Oriented Web Engineering. In *ACM SAC2000*, 2000.

[Hinz et al. 2004] Michael Hinz, Zoltán Fiala, and Frank Wehner. Personalization-Based Optimization of Web Interfaces for Mobile Devices. In *Mobile HCI 2004, Glasgow*, 2004.

[Lyman et al. 2003] Peter Lyman, Hal R. Varian, Kirsten Swearingen, Peter Charles, Nathan Good, Laheem Lamar Jordan, and Joyojeet Pal. *How much information? 2003 Executive Summary*. University of California at Berkeley, 2003.

[Murugesan and Deshpande 2001] San Murugesan and Yogesh Deshpande, editors. *Web Engineering*. Springer Verlag, LNCS Vol. 2016, 2001. ISBN: 3-540-42130-0.

[Schwabe et al. 1996] Daniel Schwabe, Gustavo Rossi, and Simone D. J. Barbosa. Systematic Hypermedia Application Design with OOHDM. In *Hypertext '96*, 1996.

[Szyperski 1998] Clemens Szyperski. *Component Software, Beyond Object-Oriented Programming*. Addison-Wesley, 1998. ISBN: 0-201-17888-5.

[Vdovjak et al. 2003] Richard Vdovjak, Flavius Frasincar, Geert-Jan Houben, and Peter Barna. Engineering Semantic Web Information Systems in Hera. *Journal of Web Engineering, Rinton Press*, 2(1&2):003–026, 2003.

[Wehner and Lorz 2001] Frank Wehner and Alexander Lorz. Developing Modular and Adaptable Courseware Using TeachML. In *ED-MEDIA, Tampere, Finland*, 2001.