Submitted to the 3rd Knowledge Engineering Forum

A Hypermedia-Based Information System for the Visualization of Complex Design Processes*

Frank Leidermann

Expert System Group, University of Kaiserslautern, P.O. Box 3049, D-67653 Kaiserslautern, Germany
<leiderma@informatik.uni-kl.de>

Abstract

This paper describes the methodical development of an information system which visualizes complex, distributed and cooperative design processes for different user groups in the World Wide Web.

1 Introduction

Traceability of complex design processes is an important but difficult goal, especially within long-term (team) projects.

Even for people directly involved in a project, it is hard to overview the internal dependencies of the process and the created products. This problem can be both, a result of and a reason for insufficient communication.

Often, outsiders, e.g. the customers of a project, must also get a general idea of the design process. Sometimes, a comprehensible and transparent procedure is even required by law, e.g. in land-use planning, the application domain of the interdisciplinary research project *"The Intelligent Land-Use Plan (IBP)"* at the University of Kaiserslautern.

A hypermedia-based *information system* can significantly increase the transparency of design processes. And if it concerns a computer-aided process, e.g. founded on a workflow-management system, the necessary data are already available.

The system *CoMo-Kit*, developed at the University of Kaiserslautern, supports complex, distributed, cooperative design processes. Within the research project *IBP* [1] the system is used as a base tool for computer-aided development of legally binding land-use plans.

CoMo-Kit was extended by a component called *Information Assistant* [2] which visualizes the design process as well as the resulting products for different user groups.

In the following, some aspects of CoMo-Kit which are relevant for this paper will be shortly described. After this, the development of the Information Assistant will be presented, regarding requirements, methodology and realization.

2 Causal Dependencies in CoMo-Kit

CoMo-Kit includes the two components *Modeler* and *Scheduler* (see figure 1).

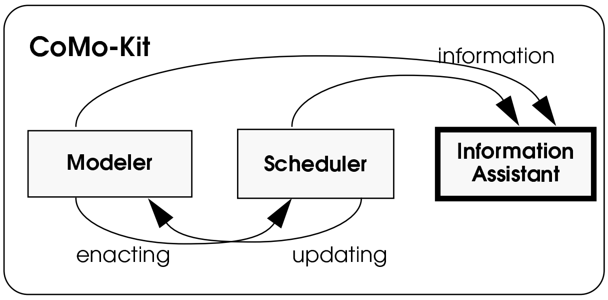


Fig. 1: The rough architecture of CoMo-Kit

The Modeler serves as a tool for the *planning* of design projects [3]. The process to be enacted, as well as the products to be created, are modeled.[1]

The *enactment* of a project is coordinated and documented by the Scheduler [4] [5].

An important task of the Scheduler is the management of the causal dependencies within complex design processes. During the enactment the dependencies must stay consistent, especially after changes.

One *example* for a causal dependency: Within a land-use plan the type of use for building is designated for each area, e.g. residential or industrial use. Designating different types of use for adjacent areas may cause conflicts. So there exists causal dependencies between these designations.

In CoMo-Kit causal dependencies are automatically derived from the present project plan. Furthermore, dependencies can be dynamically added or removed during the enactment.

*This work is part of the interdisciplinary research project *The Intelligent Land-Use Plan* at the University of Kaiserslautern. The project is supported by the *Volkswagen-S*tiftung *(I/68 707, I/69 374 - 375).*

[1] During the enactment the project plan can be refined or changed.

The causal dependencies are represented by a complex Design Rationale Network which mainly consists of (valid or rejected) design decisions, process steps and created products. The implementation of the Scheduler is build on the general planning and design model REDUX [6].

Essentially, the task of the CoMo-Kit Information Assistant is the visualization of this network.

3 The Information Assistant

3.1 Requirements

As the information to be visualized builds a complex network, the structuring and presentation by hypertext/hypermedia techniques are nearly self-evident. Additionally, the integration into the World Wide Web (WWW) allows a distributed and platform-independent access. Therefore, the Information Assistant should automatically generate a *hypertext application* consisting of HTML-documents which visualize a concrete project planned and enacted with CoMo-Kit.

A special challenge is the fact that there are different *"profiles of use"* [7] which are mainly composed of the users' tasks and qualifications (see section 3.4).

The Information Assistant must be *generic* in order to ease the adaption of the hypertext-application to varying user requirements. Therefore, beside software-ergonomic guidelines [2], a suitable design methodology is necessary.

3.2 The Design Methodology

The conceptional development of the hypertext application mainly founds on the design methodology *OOHDM*[1]. The general *object-oriented approach* of this methodology makes the design and the implementation easier since the CoMo-Kit Modeler bases on object-oriented modeling techniques and is implemented in the object-oriented language Smalltalk. Moreover, OOHDM considers that hypertext applications must often be developed for *different user groups*.

The actually used methodology distinguishes between the three design steps *Domain Analysis*, *Navigational Design* und *Presentation.*

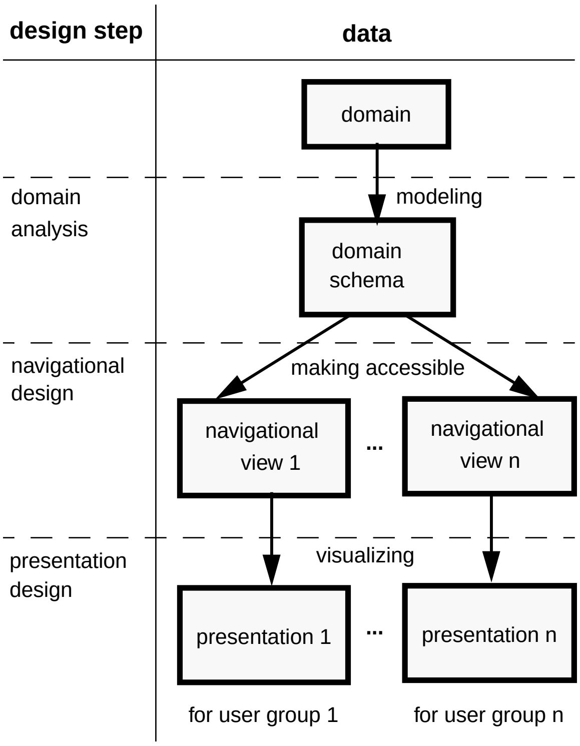


Fig. 2: The three design steps and their results

During the domain analysis the domain is *modeled* by a *domain schema.* It describes the structure of the information to be visualized. A domain schema can be seen as an Entity-Relationship-Model [9] expanded by the object-oriented abstraction concepts generalization and aggregation.

Founded on the domain schema, different *navigational views* are designed. A navigational view defines which information is *accessible* for a certain user group and in which way the information can be accessed.

The hypertext network to be generated for this group is described by classes of nodes, links and access structures, eg. indices. Nodes correspond with entities, links with relationships. Therefore, a navigational view can be directly derived from a domain schema by associating *exactly one* node (resp. link) class with each entity (resp. relationship) class.

But in most cases, a navigational view differs from the domain schema. On the one hand, information can be filtered (e.g. by 'omitting' certain entity classes). On the other hand, information can be structured in a different way. This is substantiated in section 3.4.

The way of *presentation* of the nodes, links and access structures is defined in the third design step.

[1] Object-oriented Hypermedia Model ([8])

In the following, the domain schema, the navigational views, and the presentations designed during the development of the Information Assistant will be briefly characterized. The final subsection 3.6 deals with the realization.

3.3 The Domain Schema

The domain schema developed during the analysis describes CoMoKit design processes. It concerns different topics:

Products

- which (intermediate) products have been created ?
- are these products composed of several components?

Process Steps or Tasks

- who is responsible for a certain process step?
- which tasks have not been enacted yet?
- in which way a task has been enacted?

Design Decisions

- what was the reason for a certain process step? did the accordant decision found on other design decisions or other products?
- which alternative decisions were possible?

Rejected Design Decisions

- are there any alternative decisions having been rejected? which products were affected?
- what was the reason for the rejection of alternative decisions? did the rejection found on other rejected decisions or created products?

The resulting domain schema basically contains entity classes representing products, tasks and decisions.

3.4 The Navigational Views

For the application area land-use planning two navigational views were derived from the domain schema. In an abstract way they describe the hypertext networks for two user groups with different user profiles:

The Project-Internal View

The first user group is directly involved in the project: the town planners. The entire process is presented to this group. The causal dependencies and the reasons for design decisions or changes are shown as well as the resulting products (i.e. designations). This *process-oriented* view is structural similar to the developed domain schema.

The Project-External View

The other user group includes the outsiders, e.g. interested citizens or investors. The accordant hypertext network is necessarily simpler structured and filters the information being irrelevant for this group.

This view is rather *product-oriented* since outsiders need primary information concerning certain pieces of land. Products being composed of several components are represented in a more compact (and therefore clearer) way. Nevertheless, the main causal dependencies which make the process comprehensible are also presented to this group.

This navigational view basically contains one (compact) node class corresponding with all entity classes of the domain schema.

Other Application Areas

The differentiation between these two user groups can be transfered to other application domains, e.g. *software development*: on the one side there are the *developers*, on the other side the *customers*.

Naturally the first user group could be split into further user groups, e.g *project managers* and *technical developers*.

3.5 The Presentations

The concrete nodes, links and access structures, e.g. indices, are *textually* or *graphically* presented by HTML-documents.

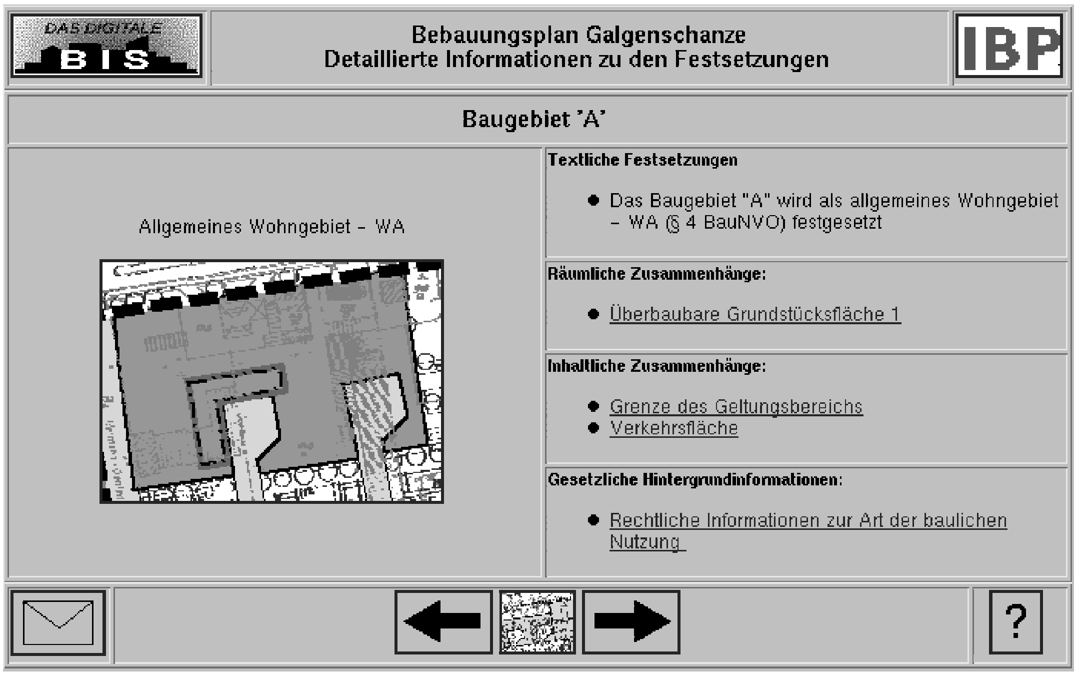
Often the resulting *products* are anyway graphically created, e.g. parts of a land-use plan. In this application domain information can be accessed via interactive drawings of the planned area (see figure 3). Even *process* steps can be partly graphically presented by an interactive picture since CoMo-Kit includes a graphical modeler tool.

3.6 The Realization

The HTML-documents are automatically *generated*. On the one hand, a document can be created as the result of an WWW-inquiry. On the other hand, all documents can be generated at one point of time as a complete stand-alone version.

Within the CoMo-Kit Modeler it is possible to define which products (or components) should be *visible* for a certain user group.

Analogous to the three steps of the design methodology the implementation of the Information Assistant consists of several *layers*. As a consequence the logical structure and the presentation of the hypertext application can be independently adapted to varying user requirements.



Bebauungsplan Galgenschanze
Detaillierte Informationen zu den Festsetzungen

Baugebiet 'A'

Allgemeines Wohngebiet - WA

Textliche Festsetzungen

- Das Baugebiet "A" wird als allgemeines Wohngebiet - WA (§ 4 BauNVO) festgesetzt

Räumliche Zusammenhänge:

- Überbaubare Grundstücksfläche 1

Inhaltliche Zusammenhänge:

- Grenze des Geltungsbereichs
- Verkehrsfläche

Gesetzliche Hintergrundinformationen:

- Rechtliche Informationen zur Art der baulichen Nutzung

Fig. 3: A HTML-document generated for the application area land-use planning (user group citizens)

4 Conclusion and Current Work

Conclusion

The Information Assistant visualizes the design process and its resulting products. The internal dependencies and the reasons for design decisions or changes are presented to different user groups.

It is obvious that the used methodology was necessary for the genericity of the Information Assistant. This is important for the easy adaption to varying user requirements.

Current Work

Within the research project "Intelligent Land-Use Plan" the Information Assistant is the kernel of an information system for citizens. Until now, structure and presentation of the HTML-documents has been improved for this user group. Currently, the accordant is aspired for the user group town planners.

References

[1] G. Pews. *Der intelligente Bebauungsplan*. URL: http://wwwagr.informatik.uni-kl.de/~pews/ibp.html, 1996.

[2] F. Leidermann. *Ein Hypermedia-basiertes Informationssystem zur Visualisierung komplexer Arbeitsabläufe am Beispiel der Bebauungsplanung*. Master Thesis, University of Kaiserslautern, 1996.

[3] F. Maurer. *Modelling the Knowledge Engineering Process*. In: F. Maurer (ed.), *2nd Knowledge Engineering Forum 96* in SFB 501 Bericht 01/96, 1996.

[4] B. Dellen, F. Maurer, J. Paulokat. *Verwaltung von Abhängigkeiten in kooperativen wissensbasierten Arbeitsabläufen*. In: M.M. Richter and F. Maurer (ed.), Proceedings in Artificial Intelligence 2, Expertensystem 95, 1995.

[5] F. Maurer and G. Pews. *Flexibles Workflowmanagement für Entwurfsprozesse am Beispiel der Bebauungsplanung*. Proceedings CAD '96, 1996.

[6] Ch. Petrie. *Planning and Replanning with Reason Maintenance*. Dissertation, University of Texas, Austin, 1991.

[7] M. Hofmann and L. Simon. *Problemlösung Hypertext*. Hanser, München, Wien, 1995.

[8] D. Schwabe and G. Rossi. *The Object-Oriented Hypermedia Design Model*. In: *Designing Hypermedia Applications*. Communications of the ACM, 38(8), August 1995.

[9] P. Chen. *The entity-relationship approach: Toward a unified view of data*. ACM Transactions on Data Base Systems, 1(1):9-36, 1976.