# Adaptable Hypermedia with Web Standards and Tools

Lloyd Rutledge, Lynda Hardman, Jacco van Ossenbruggen and Dick C.A. Bulterman
CWI
P.O. Box 94079
1090 GB Amsterdam, The Netherlands
E-mail: {lloyd,lynda,jrvosse,dcab}@cwi.nl

The World Wide Web offers the promise, and more and more the reality, of a widely distributed and widely accessible digital library of related hypermedia data. In order for this data to be stored, accessed and played it should not be encoded as final form presentations, since these consume storage space and cannot easily be adapted to variations in presentation-time circumstances such as user characteristics, changes in end-user technology and the interactions of the user at presentation time. Instead, a more presentation independent approach needs to be taken that allows the dynamic generation of presentations that are derived from a presentation-independent description and are adapted to run-time circumstances.

A collection of standards exists that address this possibility and its related challenges. The ISO standard HyTime (Hypermedia/Time-based Structuring Language) [10] specifies the representation of hypermedia documents in a presentation independent format. HyTime is used to a small but steady degree by private industry for large-scale text document collections. The widespread adoption of HyTime is inhibited because there are few widely available tools for processing it, there are few examples of its use, and few widely available systems exist that can generate hypermedia presentations from it. The ISO standard DSSSL (Document Style Semantics and Specification Language) [11] defines the transformation of electronic documents into formats that present them. SMIL (Synchronized Multimedia Markup Language, pronounced "smile") [9] is a new W3C (World Wide Web Consortium) recommendation for immediately presentable hypermedia documents distributed on the World Wide Web. DSSSL transforms documents encoded with Standard Generalized Markup Language (SGML), which is used as the foundation for defining both HyTime and SMIL. Because of this, DSSSL can encode the transformation of documents from HyTime to SMIL, and thus can encode the final presentation of documents stored in HyTime. The use of DSSSL with HyTime was recently made easier with the release of the second edition of HyTime, which contains new facilities for use with DSSSL. Publicly available tools exist that make the cooperative use of HyTime, DSSSL and SMIL for hypermedia digital libraries widely implementable. These standards and tools can be used together to create an environment that processes documents from stored hypermedia data into final presentations.

In this presentation we present the Berlage environment design [15], [16], which applies these public standards and tools to create a storage-to-presentation hypermedia system.The design of Berlage is based on the Standard Reference Model for Intelligent Multimedia Presentation Systems (SRM-IMMPSs) [2], [15], which defines how the specification of automatically generated dynamic tailored presentations can be divided up into distinct modules. This dicussion of Berlage and the SRM-IMMPSs is illustrated with an example application about the city of Amsterdam, The Netherlands, called Fiets (Foundation for Interactive Electronic Touring Systems, or fiets {pronounced "feets"} [14], the Dutch word for "bicycle" and generally the preferred means of personal transportation in Amsterdam). Fiets provides a hypermedia interface to a digital library of media data regarding the city of Amsterdam, The Netherlands.

## The Fiets Hypermedia Application

Hypermedia has typically been modeled in terms of its presentation. The resulting structure usually represents the timing of the media object presentations, the spatial layout of the visual media object displays, and the point-and-click navigational interface for the user. However, there is often a discernible structure to the concepts being conveyed by the presentation, and there can often be more than one means of using hypermedia presentation structure to convey these concepts and their underlying structure. Because of this, it is often beneficial for an author to model a document in terms of such an underlying conceptual structure and handle the mapping of this structure to the final presentation as a separate task.

The Fiets application is used to explore the differences between time, space and hyperlinks in storage and presentation. The temporal structure of a document may have a direct mapping to the temporal structure of its presentation, but this is not always so. It could instead map to spatial or navigational presentation structure. Also, its mapping to the presentation structure may be altogether indirect or nonexistent. It is hoped that by exploring the mappings among these corresponding structures that the distinction between storage and presentation will be better understood and the potentials of maintaining such a distinction better utilized.

The Fiets application is used as an example to illustrate these structural distinctions. Fiets consists of media objects about Amsterdam and metadata about these objects. These media objects consist of photographs of historic buildings along the Herengracht, one of Amsterdam's main canals. Fiets conveys the underlying conceptual structure of the Herengracht. The Herengracht has geographic structure: its buildings have locations, represented as street addresses. With its history, the Herengracht has a temporal structure: its buildings were constructed on particular dates. Further, the Herengracht buildings have structure consisting of semantic relationships. In Fiets, this is exemplified with detail images of building exteriors, which show particular portions of each building.

Fiets illustrated the distinction between the time, space and hyperlinks of storage and of presentation by defining 9 different mappings of the possible combinations of each. These mappings are illustrated in Figure 1. Figure 2 shows a screen display from the generation from one such mapping: the "panorama" mapping of the spatial structure of the storage media and metadata (the street) to the spatial layout of the presentation (the screen display).

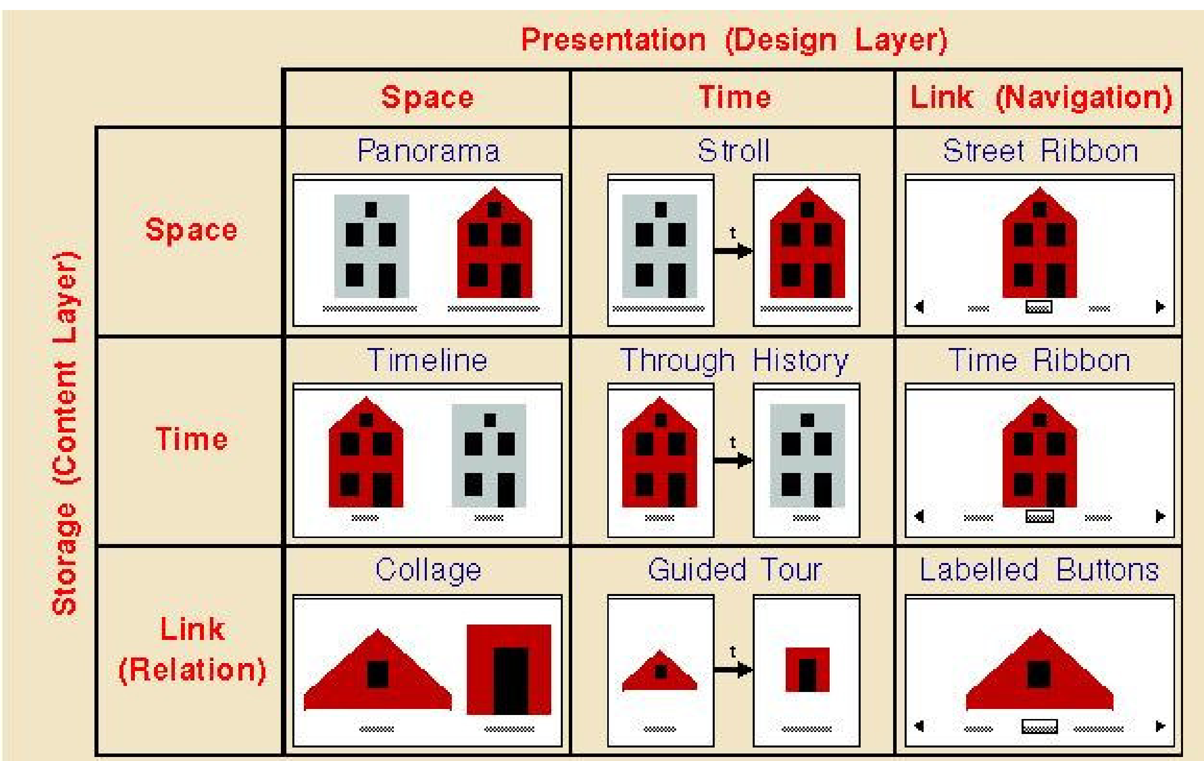

**Figure 1**


**Figure 2**

# The SRM-IMMPSs

The SRM-IMMPSs is illustrated in Figure 3 and described below.

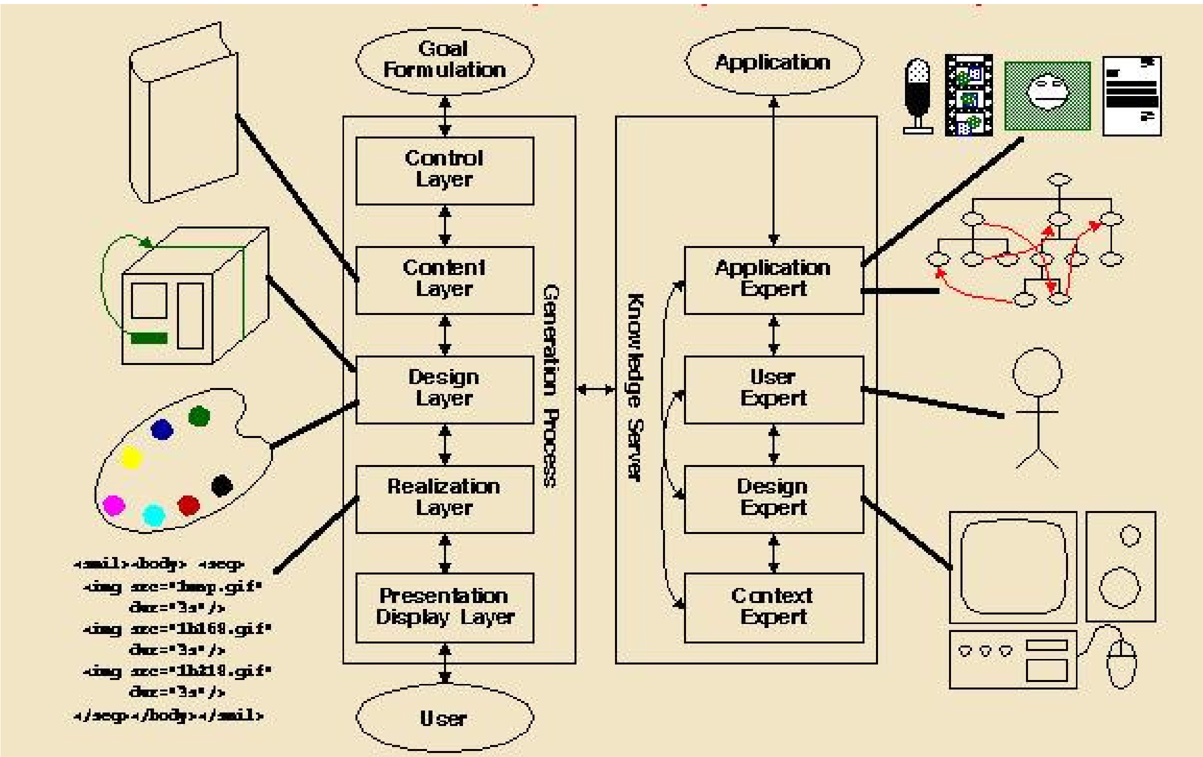


**Figure 3**

The goal formulation component of the SRM-IMMPSs handles the initial interaction with the user that starts the presentation and establishes its goals. These goals are then passed to the control layer, which processes the goals to make sure they are met. While the breaking up of each goal into subgoals is determined by the content layer, the control layer is in charge of processing these subgoals to determine which is to be met next. This chosen subgoal is then communicated to the content layer, which generates the portion of the presentation that achieves it.

In a paper-based society, a person reads a document to understand a particular body of knowledge. He or she can expect to be able to communicate about this topic with other people who have read the same paper document. The same common understanding should be expected for two people who have been presented with the same hypermedia document, no matter how much the document may have been adapted differently for each user. The content layer ensures that the recipient of the presentation for a given document has been properly shown all information within that document, regardless of how the other layers adapt the presentation.

The design layer of the SRM-IMMPSs determines what means the final presentation uses to meet the goals established in the content layer. This layer makes decisions on the "look and feel" of the presentation, including what media objects should be selected and how they should be presented. The design layer also determines the spatial–temporal and navigational layout of the final presentation. The different style combinations of storage and presentation structure demonstrated by Fiets are shown in Figure 1.

The realization layer translates the desires expressed by the design layer into a directly playable hypermedia format. This layer determines exactly what spatial coordinates and what timing can match the constraints given by the design layer. If no such detailed specifications are possible, it then communicates with the design layer to determine an acceptable alternative. Similar communication occurs when the initial desires of the design layer cannot be expressed in the directly playable hypermedia format. The two layers then decide what acceptable alternative can be expressed in the output format. In Fiets, the realization layer corresponds with the generation of the SMIL code that makes up the final presentation to the user.

The presentation display layer is embodied by the particular player software that is called at presentation time to process a segment of code generated by the realization layer for presentation to the user. It also handles the user's interaction with the generated presentation. Because the presentation is fixed at this point, there are no style modules for this layer.

The application expert provides an interface for the various systems and formats that are involved in generating the presentation. It can provide the generation process information about what is encoded by some input of a particular format. It can also generate output encoded in a particular format as instructed by the generation process. Such a format can be for the media type of a media object to be included in the final presentation. A separate application expert module could exist for the input and output of each format used in a generated presentation. With this model, any number of application expert modules could be used simultaneously for the generation of a single presentation.

The application expert can also provide access to the format in which archival hypermedia information representing more general concepts and facts is stored. This represents the knowledge of a particular semantic domain. The application expert provides the generation process with access to this information and enables it to be transformed into presentation to the user.

The user expert provides information about the user that is processed to adapt the presentation to the user's characteristics. These characteristics include abilities, preferences and areas and levels of expertise. A style module for this layer would encode this information about a user. One style of presentation can be tailored for different users by switching only the user expert modules. One user could have such a module that could be plugged into any presentation generating style.

The design expert communicates the constraints of the final presentation environment. This includes primarily information about the platform on which the presentation is rendered, such as what media peripherals are available.

The context expert keeps track of the activity that has occurred during a given presentation. This information is important to the content layer in determining whether goals and subgoals have been met and in determining what future actions are required to complete them. There is no pre-written module required for this expert because the information it provides is generated automatically during the presentation itself. Also, the processing of this information is specified for the most part by the content layer module.

# The Berlage Environment Design

The current design of the Berlage architecture is illustrated in Figure 4. Its components are each briefly described below.

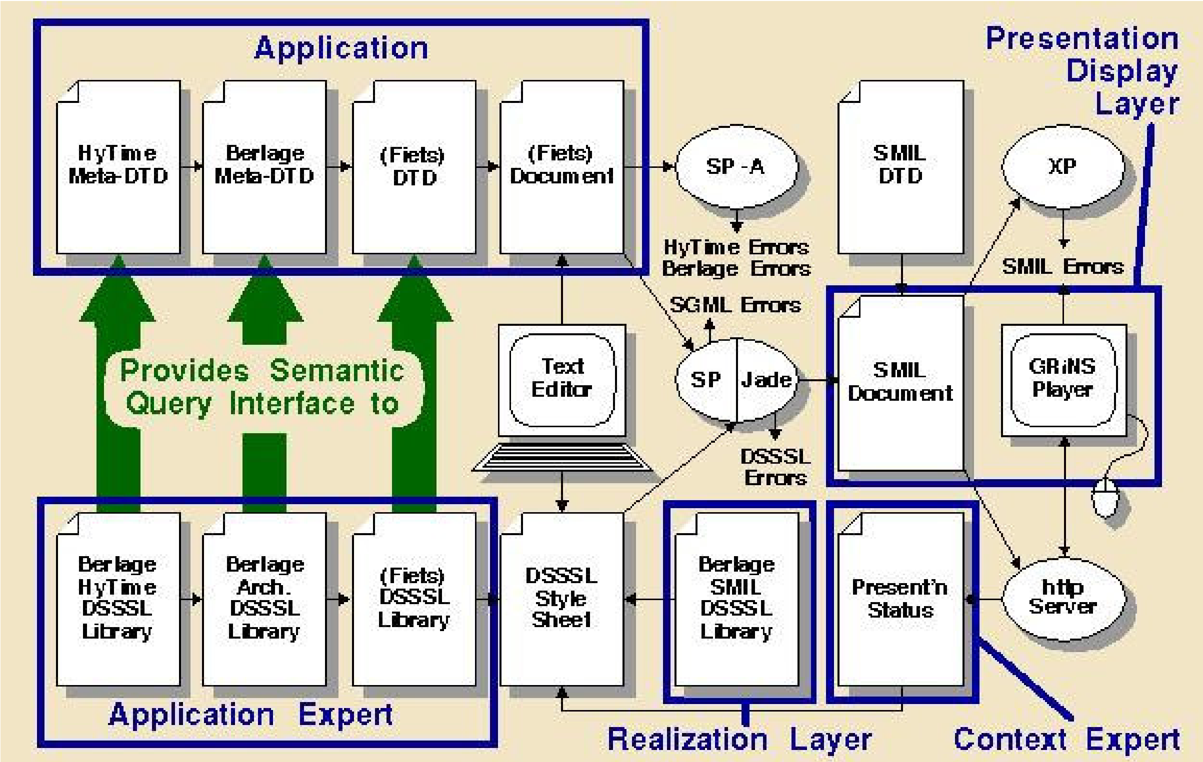


**Figure 4**

HyTime is used to encode the presentation-independent hypermedia semantics of the stored documents [10]. HyTime is a syntactic subset of SGML, [12] extending the semantics SGML encodes into hypermedia. HyTime syntax is defined as an SGML architecture with a meta-DTD [10]. The semantics encoded by HyTime constructs in a document can be queried for using the property sets defined in the HyTime standard. HyTime was used in Fiets to represent such relatively common hypermedia semantics as a year in history.

An SGML architecture defines a broad set of SGML and HyTime documents that share semantics within a particular conceptual domain. Its syntax is defined with a meta-DTD. Access to the semantics represented by its syntax is defined with property sets. SGML architectures can inherit from one or more other architectures, and inherit their property sets. An SGML architecture called Berlage is defined for the Berlage environment [15]. It includes such common hypermedia semantics as dimensions in time or pixels of a media object.

A DTD more narrowly defines the syntax of a document set. Property sets can be defined for DTDs as with architectures. This syntax of the Fiets document collection about Amsterdam is defined as a DTD. An individual document can be validated in terms of all levels of syntax described above with the tool SP [6].

DSSSL is a lisp dialect encoding the transformation of SGML and HyTime documents into other SGML documents that are typically directly processed for presentation [11]. DSSSL programs are called style sheets, providing the basis for the term "style" as used in this paper. DSSSL provides inclusion mechanisms that enable the division of DSSSL code into libraries shown in Figure 3. DSSSL can query against HyTime-defined properties. Some of the DSSSL libraries in Fiets define the functions that process these properties, a technique which is described in the initial work on the Berlage environment [16]. Jade is a publicly available DSSSL engine [5].

SMIL is an XML-compliant, HTML-like W3C recommendation for hypermedia on the Web [9]. SMIL is easily processed as output by DSSSL because it is encoded as XML , which is a subset of SGML. XP is an XML parser which is used here to validate SMIL code [7]. Berlage provides DSSSL functions to facilitate the generation of SMIL output. GRiNS is a publicly available player for SMIL presentations [4], [8].

Part of the incorporation of the SRM-IMMPSs into Berlage involved introducing dynamics [15]. This is done with an http server that outputs DSSSL encoding the user interaction history of the presentation. This DSSSL code representing the presentation status is then incorporated into the style sheet for processing the next step of the presentation.

The style modules from the mix'n'match scheme would be represented in the Berlage environment as components of the main DSSSL style sheet. The entire style sheet should be divisible into the modules described below. As DSSSL code, the style sheet code would thus consist of an inclusion statement for each module used. These DSSSL style module files can be stored locally or accessed from anywhere on the Web with URLs and Jade's processing of them. This enables the distribution of style as well as content.

# Conclusion

This presentation discusses dynamic adaptable hypermedia storage and presentation. The SRM-IMMPSs is presented as a model on which to base the implementation of dynamic adaptable hypermedia. The Berlage environment is presented as an implementation architecture for hypermedia environments based on SRM-IMMPSs that use current public standards and tools. Fiets is presented as an illustrative example of SRM-IMMPSs and the Berlage environment.

# References

1. City of Amsterdam Municipal Department for Preservation and Restoration of Historic Buildings and Sites. Amsterdam Heritage, URL: http://www.amsterdam.nl/bmz/adam/adam.html.
2. Bordegoni, M., Faconti, G., Feiner S., Maybury, M.T., Rist, T., Ruggieri, S., Trahanias, P. and Wilson, M. A. Standard Reference Model for Intelligent Multimedia Presentation Systems. Computer Standards and Interfaces 18(6,7) (December 1997), pp. 477-496.
3. Bray, T., Paoli, J. and Sperberg-McQueen, C.M., Extensible Markup Language (XML), W3C Recommendation (January 1998), URL: http://www.w3.org/TR/REC-xml.html.
4. Bulterman, D.C.A, Hardman, L., Jansen, J. Mullender, K.S. and Rutledge, L. GRiNS: A GRaphical INterface for Creating and Playing SMIL Documents, in Proc. Seventh International World Wide Web Conference (Melbourne, Australia, April 1998).
5. Clark, J. Jade -- James' DSSSL Engine, URL: http://www.jclark.com/jade/.
6. Clark, J. SP -- An SGML System Conforming to International Standard ISO 8879 -- Standard Generalized Markup Language, URL: http://www.jclark.com/sp/.
7. Clark, J. XP -- An XML Parser in Java, URL: http://www.jclark.com/xml/xp/.
8. CWI (Centrum voor Wiskunde en Informatica). GR i NS -- The GRaphical i Nterface to SMIL, URL: http://www.cwi.nl/GRiNS/.
9. Hoschka, P. (ed.). Synchronized Multimedia Integration Language, World Wide Web Consortium Recommendation. June 1998. URL: http://www.w3.org/TR/REC-smil/.
10. International Standards Organization. Hypermedia/Time-based Structuring Language (HyTime), Second Edition, ISO/IEC IS 10744:1997. International Standards Organization, 1997.
11. International Standards Organization. Document Style Semantics and Specification Language (DSSSL), ISO/IEC IS 10179:1996. International Standards Organization, 1996.
12. International Standards Organization. Standard Generalized Markup Language (SGML), ISO/IEC IS 8879:1985. International Standards Organization, 1985.
13. International Standards Organization. Coding of multimedia and hypermedia information -- Part 5: Support for base-level interactive applications (MHEG-5), ISO/IEC IS 13522-5:1997. International Standards Organization, 1997.
14. Rutledge, L. Hardman, L., van Ossenbruggen, J. and Bulterman, D.C.A. Structural Distinctions Between Hypermedia Storage and Presentation, in Proc. ACM Multimedia 98 (Bristol, England,

September 12-16, 1998), pp. 145-150.
15. Rutledge, L., Hardman, L., van Ossenbruggen, J. and Bulterman, D.C.A. Implementing Adaptability in the Standard Reference Model for Intelligent Multimedia Presentation Systems, in Proc. Multimedia Modeling 98 (Lausanne, Switzerland, October 12-15, 1998).
16. Rutledge, L., van Ossenbruggen, J., Hardman, L. and Bulterman, D. Practical Application of Existing Hypermedia Standards and Tools, in Proc. Digital Libraries 98 (Pittsburgh, USA, June 23-26, 1998), pp. 191-199.

**active web home** || **papers** || **posters**