Practical Application of Existing Hypermedia Standards and Tools

Lloyd Rutledge, Jacco van Ossenbruggen, Lynda Hardman and Dick C.A. Bulterman*

CWI
P.O. Box 94079
1090 GB Amsterdam, The Netherlands
E-mail: {lloyd,lynda,dcab}@cwi.nl

*Vrije Universiteit
Dept. of Math. and Computer Science
De Boelelaan 1081
1081 Amsterdam, The Netherlands
E-mail: jrvosse@cs.vu.nl

ABSTRACT
In order for multimedia presentations to be stored, accessed and played from a large library they 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 and changes in end-user technology. Instead, a more presentation independent approach needs to be taken that allows the generation of multiple versions of a presentation based on a presentation-independent description.

In order for such a generated presentation to be widely viewable, it must be in a format that is widely implemented and adopted. Such a format for hypermedia presentations does not yet exist. However, the recent release of SMIL, whose creation and promotion is managed by the World Wide Web Consortium, promises to become such a format in the short term and be for hypermedia what HTML is for hypertext.

The technology for enabling this presentation-independent approach is already available, but requires the use of large and unapproachable standards, such as DSSSL and HyTime. In this paper we show that these two standards can be used with SMIL, and by concentrating on a particular application, illustrate the use of publicly available tools to support the generation of multiple presentations from a single presentation-independent source.

KEYWORDS: Hypermedia, HyTime, DSSSL, SMIL, SP, Jade, Berlage, GRiNS.

INTRODUCTION

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 applicable to a wide variety of presentation styles, circumstances and future technology changes, it must be represented in a presentation independent format that does not prescribe aspects of its presentation. In order for such data to be presented, a mechanism must exist that can process it to generate a presentation appropriate for a given situation.

A collection of standards exists that address this possibility and its related challenges. The ISO standard HyTime (Hypermedia/Time-based Structuring Language) [8][5] 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 wide-spread 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") [7] 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) [12][6], 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.

Public domain tools exist that make the cooperative use of HyTime, DSSSL and SMIL for hypermedia digital libraries widely implementable. SP [4] is a public domain tool that can parse and validate HyTime documents. Jade (James' DSSSL Engine) [3] is a public domain tool for processing DSSSL style sheets. We have developed a public domain tool that plays SMIL documents called GRiNS (A GRaphical INterface for creating and playing SMIL documents) [2]. These standards and tools can be used together to create an environment that processes documents from stored hypermedia data into final presentations.

In this paper we present a hypermedia application about the city of Amsterdam, The Netherlands, called Fiets (Foundation for Interactive Electronic Touring Systems, or *fiets* {pronounced "feets"}, the Dutch word for "bicycle" and generally the preferred means of personal transportation in Amsterdam). Fiets is an application of the standards HyTime, DSSSL and SMIL and the tools SP, Jade and GRiNS. It provides a hypermedia interface to a digital library of media data regarding the city of Amsterdam, The Netherlands. Fiets uses code from the Berlage project, which is described in earlier work [13][14][15]. We have used public domain tools for Fiets to demonstrate how digital libraries such as the one accessed by Fiets can be readily implemented.

First we introduce the standards and tools involved. Following this is a description of Fiets and how it uses these standards. This description is illustrated with a simple example. We end this paper by exploring how these standards and tools can address these issues in a broader context.

STANDARDS AND TOOLS

In this section we introduce the standards and tools used in this paper.

HyTime

HyTime is an ISO standard for representing presentation independent hypermedia data. It is built upon SGML, which provides the basic structuring information that applies to document data in general. HyTime adds more complex structuring constructs and attaches hypermedia semantics to certain patterns of composites of this structure. The basic hypermedia semantics that HyTime represents include *hyperlinking*, which establishes descriptive relationships between document objects, and *scheduling*, which puts document objects in coordinate systems that can represent spatial and temporal relationships.

HyTime and SGML are generally considered to encode documents that are presentation independent. They can apply to a wide variety of presentation situations but do not themselves represent particular presentations. HyTime and SGML documents typically must be processed into a different format appropriate for final presentation.

HyTime and SGML are meta-languages. They encode not only individual documents but also the document sets to which they belong. A document set is defined by an SGML *document type definition (DTD)*. An individual document conforms to a particular DTD. A DTD defines a specific syntax, in terms of SGML constructs, that its documents must follow. HyTime inherits from SGML the use of DTDs to define individual document sets.

A DTD is one way to define a set of SGML documents in terms of restricted SGML syntax. Another way, defined in the HyTime second edition, is with SGML *architectures*. One difference between DTDs and architectures is that architectures define a looser, broader syntax. Multiple document sets, as defined by DTDs, can conform to a single architecture. Further, architectures can inherit from other architectures. An SGML architecture is defined by a piece of code called a *meta-DTD*. A meta-DTD specifies what composites of constructs from SGML, this architecture and other architectures make up each of this architecture's constructs. HyTime itself is an architecture and is defined with a meta-DTD.

HyTime also provides for defining *properties*, which apply semantic labels to composites of SGML and HyTime constructs. This facilitates the querying of the data represented in a document, and thus it facilitates the processing of HyTime documents into documents of other formats, including those for presentation. Typically, many different constructs can be used to represent the same property. In such cases, it is easier to query by the property than by all the possible different combinations of constructs that can represent that same property. HyTime constructs for property defining can be used for individual documents, for documents sets, and for architectures. The HyTime property set facility was extended in the second edition of HyTime to be more readily processed by DSSSL style sheets.

SP

SP is a public domain SGML parser. It can parse and validate any SGML document. It also produces error messages describing how a document fails to meet the SGML syntax. A document is parsed by SP along with its DTD. Any failure of the document to meet the DTD's syntax is reported.

Since HyTime documents are SGML documents, SP can be used to parse HyTime documents and report any errors in the general SGML syntax or in the syntax specified by the DTD. HyTime syntax errors can be detected with the use of the HyTime meta-DTD [9], made available with HyTime's second edition in August 1997. The HyTime meta-DTD can be processed by SP with the "A" (architecture) option, a new feature that allows syntax checking against SGML architectures such as HyTime. SP can also be used with this option to check that documents meet the syntax requirements specified for the Berlage architecture by its meta-DTD.

DSSSL

DSSSL is a Scheme dialect that describes how an SGML document is transformed into another SGML document or into a non-SGML format. Because HyTime documents are SGML documents, any HyTime document can be transformed by DSSSL. A DSSSL program is typically called a *style sheet*. The separation of style from structure and from content enforced with the distinction between DSSSL and HyTime facilitates the creation of particular styles by the author that can be applied to documents of the same document set.

The design of typical DSSSL usage is shown in Figure 1. This diagram shows how an SGML document is processed with an accompanying style sheet for that document by a DSSSL engine. The DSSSL engine determines the mapping encoded by the style sheet and generates the appropriate transformation of the source document into the presentation format.

DSSSL is designed to work with HyTime-defined properties, as specified in the HyTime second edition. A style sheet can ask for the property of a document object with one function call, rather than requiring a complex section of code that checks for all the syntax composites that could define that property. No mechanism for recognizing these properties is specified. One possibility for recognizing properties and determining their values is that DSSSL functions be defined that provide access to these properties.

Jade

Jade is a public domain DSSSL engine. A DSSSL engine accepts an SGML or HyTime document along with a DSSSL style sheet and generates a transformation of that document. Jade works with the SP SGML parser. SP parses the SGML code of a document and provides Jade with information about the SGML defined structure. Jade processes the style sheet and uses it to map this information from SP into the output document. Any errors in the SGML syntax are reported to the user by SP. Any errors in the DSSSL syntax are reported to the user by Jade.

SMIL

SMIL, a recent W3C proposed recommendation, is a format representing hypermedia presentations on the Web. It incorporates basic hypermedia principles such as spatial layout, temporal composition, synchronization and navigational hyperlinking. It also has constructs that adapt presentations to the characteristics of individual environments and users. SMIL specifies the display of multiple media items in a coordinated fashion: displaying visual items and related locations on the screen, and synchronizing the timing of the presentation of these media

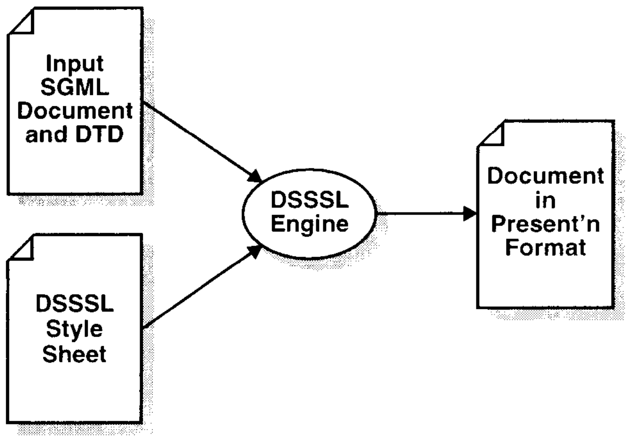

Figure 1: Typical DSSSL Usage

items. SMIL specifies the display of navigational interface allows the user to select what portions of the presentation to currently display. What is novel about the use of SMIL is its promise of wide implementation and adoption, making SMIL documents viewable by a large audience.

SMIL has an easy-to-author format whose syntax resembles HTML. SMIL is defined using XML (Extensible Markup Language), a recent W3C recommendation [1]. XML is a simplified dialect of SGML. SMIL syntax is defined with an XML DTD [21], just as HTML syntax is defined with an SGML DTD. Since SMIL documents are XML documents, they are SGML documents, and thus are easily processed as output, and as input, of DSSSL transformations.

GRiNS

GRiNS is a SMIL player and authoring environment developed at CWI. Public domain player-only versions of GRiNS for Sun, SGI, and Windows-95/NT environments are available. The GRiNS environment was adapted for SMIL from CMIFed, a research hypermedia environment that uses similar structures in its format, CMIF [20].

THE FIETS DOCUMENT SET AND PROCESSING ENVIRONMENT

Fiets is a hypermedia application that stores and presents documents about Amsterdam. It uses the standards and tools just described in a manner illustrated in Figure 2 and described below.

HyTime is used by Fiets to represent the location on the Web of a large number of media objects regarding Amsterdam. These media objects include images such as paintings by Dutch masters, photographs of modern Amsterdam buildings and neighborhoods, old paintings and engravings of buildings and neighborhoods as they once appeared, videos and sound clips of modern Amsterdam life, and text descriptions of each of these. HyTime also encodes information about what it is in the real world that these media objects represent or relate to. Such information includes a painting's painter, title, and year of creation, the address of a building in a photograph with its architect and year of construction, and who is represented in a painting or picture, along with their names, life spans, and addresses when living in Amsterdam.

SMIL is used to present Fiets documents to the user in a typical hypermedia fashion. An advantage of using SMIL in the Fiets environment is that it is an SGML subset and can thus be processed using DSSSL. The use of DSSSL provides Fiets authors with the ability to more quickly generate a wide variety of presentations of the same source document. A large document can be written once for long-term storage, and it can be presented in many different ways without re-editing the original document. With the document content and general structure being established by HyTime, the author of DSSSL style sheets can focus on the desired style of presentation, the mapping of the navigational interface, and the adapting of the presentation to particular environments and individual users.

Standard text editors are used to create Fiets documents and style sheets. Fiets documents are run through SP to validate them or to provide error messages to assist the author in checking the document's SGML and HyTime conformance. A Fiets document and corresponding style sheet are input to Jade and a SMIL document is generated. This SMIL document is then presentable across the Web with GRiNS players.

Both Fiets documents and the style sheets that transform them include code from the Berlage project (named after H.P. Berlage, the architect of several important buildings in Amsterdam), which provides code that is applicable to hypermedia document collections in general, not just Fiets. This code includes the meta-DTD specifying the Berlage architecture and the DSSSL code making up the Berlage libraries.

Individual components of the Fiets environment are described in more detail below.

Text Editor

An author uses a text editor to create both Fiets documents and the style sheets that encode their presentation. If emacs is used, writing the Fiets document is made easier with PSGML [16], which defines some editing functions for SGML documents. If the Fiets document contains any errors, these are corrected using the text editor. Errors in the SGML syntax can be detected by processing the Fiets document with SP, either by itself or as part of Jade Similarly, the document can be processed by SP alone with the "A" option to check for any HyTime syntax errors. The SP "A" option can also be used to check the Fiets document against the Berlage architecture to check for any Berlage syntax errors. Finally, running the style sheet through Jade will generate messages about any DSSSL errors that the author must correct.

Fiets Document

Once written and cleared of any SGML, Berlage and HyTime syntax errors, a Fiets document is ready for processing by SP and Jade, along with an appropriate style sheet, into a final presentation. A Fiets document includes by reference the SGML and HyTime code making up the Fiets DTD, the Berlage meta-DTD, and the HyTime meta-DTD. This inclusion enables SP to check the syntax of the document in terms of SGML, Berlage, and HyTime.

Fiets DTD

The Fiets DTD described the syntax, in terms of SGML constructs, that can exist in Fiets documents. As is typical with DTDs for particular document sets, the Fiets DTD is fixed and cannot be modified by the authors of individual documents. This ensures that the documents created can be processed by the same style sheets. The DTD, in conjunction with the two meta-DTDs, also specifies syntax in terms of Berlage and HyTime constructs, thus restricting the Berlage and HyTime construct-based syntax of Fiets documents. The Fiets DTD contains code specifying that its documents conform to the Berlage and HyTime architectures. The Fiets DTD also defines properties that Fiets-defined objects have. This enables DSSSL style sheet code to refer to these properties. The conformance of a Fiets document to the Fiets syntax can be checked with SP.

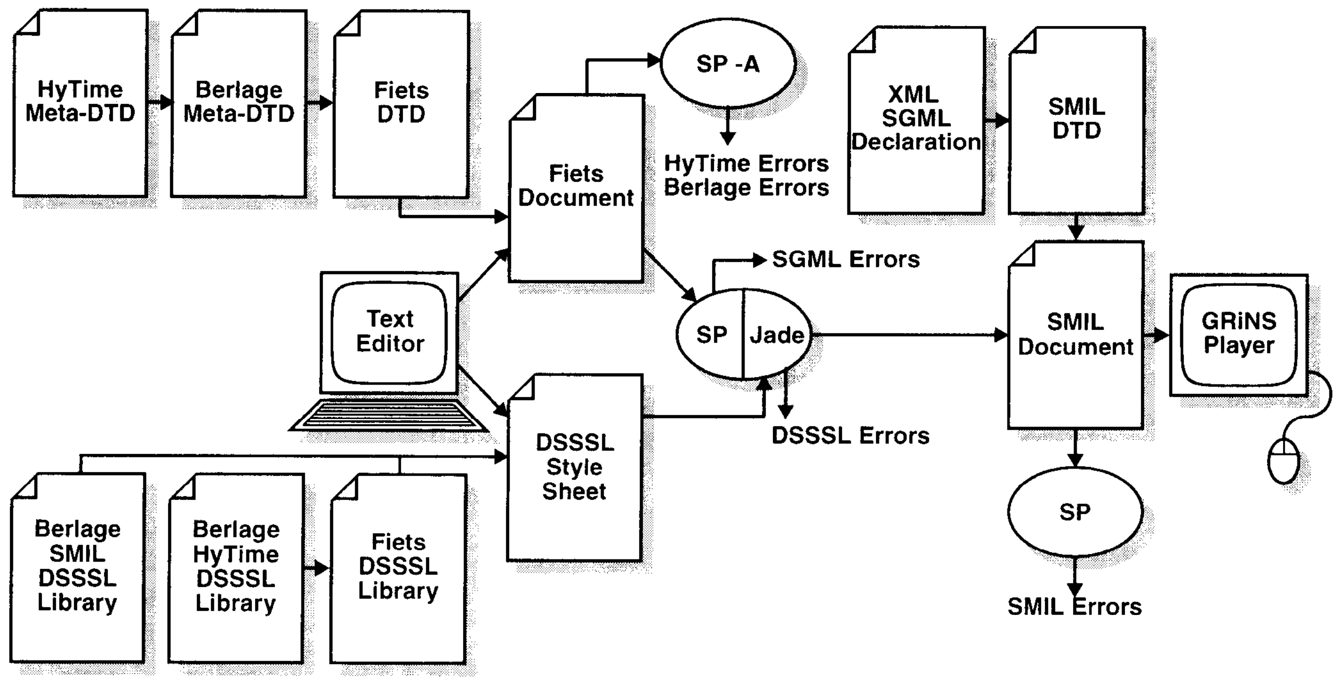

Figure 2: Fiets Environment Design

Berlage Meta-DTD

The Berlage architecture [13][14][15], specifies certain hypermedia semantics, and the SGML- and HyTime-based constructs for representing them, that apply to hypermedia document sets in general but are not represented with HyTime alone. The Berlage meta-DTD also defines properties for use with DSSSL style sheets. The Berlage architecture contains constructs, properties and associated semantics that are useful to Fiets but are more generally applicable to other documents sets as well. Many of these Berlage constructs are extensions of HyTime constructs, and their semantics fine-tune the broader semantics specified by their associated HyTime constructs. The Berlage meta-DTD contains code specifying that it conforms to the HyTime architecture. As such, it inherits from the HyTime architecture.

The conformance of a Fiets document to the Berlage syntax can be checked with SP using the "A" option and specifying the Berlage architecture as the architecture to be checked against.

HyTime Meta-DTD

The HyTime meta-DTD defining the syntax of conforming HyTime documents is available on the web [9]. It also defines properties for use in DSSSL style sheets referring to HyTime documents and their constructs. This meta-DTD is referenced by the SGML code in both the Fiets DTD and the Berlage meta-DTD. The conformance of a Fiets document to the HyTime syntax can be checked with SP using the "A" option and specifying the HyTime architecture as the architecture to be checked against.

DSSSL Style Sheet

Once written and cleared of any DSSSL errors, a style sheet for Fiets documents can be processed by Jade with a Fiets document into a SMIL presentation. Every style sheet used in the Fiets environment includes, through use of the SGML external entity mechanism, the Fiets DSSSL library and the Berlage SMIL DSSSL library. This enables the style sheets to use the specialized functions that are defined in the libraries. It also enables the style sheets to refer to and process the properties defined in the Fiets DTD and in the architectures.

Fiets DSSSL Library

The Fiets DSSSL library contains definitions for DSSSL functions that access the structure and semantics defined in Fiets documents. The library provides access to the Fiets document structure with functions that query documents based on constructs defined in the Fiets DTD. Access to Fiets document semantics is provided with functions that query on the properties defined in the Fiets DTD. This enables style sheets for Fiets documents to refer to these properties by name.

The Berlage HyTime DSSSL Library

The HyTime standard defines properties shared by all HyTime documents but does not define any mechanism for recognizing them. The Berlage meta-DTD also defines properties for the Berlage extension of the HyTime architecture. This paper introduces the Berlage HyTime DSSSL library, which defines a collection of DSSSL functions that recognize these properties. The functions for HyTime properties can be used for any HyTime document, not just the ones conforming to the Fiets DTD or the Berlage architecture. Similarly, the functions for Berlage properties can also be used for processing Berlage-conforming documents that are not in the Fiets document set.

The Berlage SMIL DSSSL Library

The Berlage SMIL DSSSL library defines functions for generating SMIL output. The Berlage SMIL DSSSL library is not part of the Berlage HyTime DSSSL library. It's external entity inclusion is not in either of the other libraries but in the style sheet itself.

SMIL Document

When given a Fiets document and a corresponding DSSSL style sheet, Jade outputs a SMIL document. This document should be immediately viewable with GRiNS or any other SMIL player. It is possible that SMIL syntax errors were generated as a result of faulty code in the style sheet. Such errors can be checked with the SP parser.

SMIL DTD

The SMIL DTD is not needed by GRiNS or any other SMIL player to display a SMIL document. As a syntax specification, however, it can be used to check for syntax errors. A SMIL document generated by Jade in this environment can be parsed by SP with the SMIL DTD to generate messages about any SMIL syntax errors that faults in the style sheet may have generated.

XML SGML Declaration

In SGML a declaration is a section of code that can be parsed with a document, its DTD and any accompanying meta-DTDs. It specifies some of the most fundamental SGML syntax constructs. An SGML declaration has been written for XML. When parsed by an SGML parser such as SP along with an XML document, the XML SGML declaration makes that document's XML syntax able to be processed as SGML. In the Fiets environment, the XML SGML declaration is needed to use SP to process a SMIL document with the SMIL DTD and generate SMIL syntax error messages. The term "XML SGML declaration" should not be confused with the term "XML declaration", which is the portion of code that appears at the beginning of an XML document.

SP

The primary function of SP in the Fiets environment is, in conjunction with Jade, to process the SGML documents that Jade then maps against style sheets to generate SMIL presentations. SP is also used to assist the author in generating document code by detecting syntax errors. Through the use of the DTDs and meta-DTDs available to the environment, SP can check for errors in SGML, Berlage, HyTime, and SMIL syntax.

Jade

The primary function of Jade in the Fiets environment is to generate SMIL documents given Fiets documents and style sheets. Jade also assists the style sheet author by generating messages about DSSSL errors that occur in a style sheet. Since Jade calls SP, running Jade with a Fiets document also generates SGML syntax error messages for that document.

GRiNS Player

The GRiNS player enables the author's audience to view the documents he or she generated with the Fiets environment. GRiNS can also be used directly by the author to check the appearance of the documents generated and to guide the author in writing appropriate Fiets documents and style sheets.

A SIMPLE EXAMPLE OF A FIETS DOCUMENT PRESENTATION

This section presents the document and style sheet code for a simple Fiets presentation. The intention is that this code can be used by implementers to more quickly create HyTime- and DSSSL-based hypermedia environments. With this example, the final SMIL presentation shows the picture of three houses on the Herengracht, a canal in Amsterdam. Each picture is shown for 2 seconds. The houses are shown in order of street number.

Fiets DTD

The portions of code from the Fiets DTD that are relevant to this example are shown in Figure 3. This code specifies the use of HyTime constructs to represent the buildings on a single street. Documents conforming to this DTD associate a street number with an image of the building located at that address. The *fiets* element, conforming to the *HyDoc (HyTime document)* HyTime construct conveys that this document is encoded in HyTime and is processable as such. Its only child element is a *streetfcs* element, conforming to the *fcs (finite coordinate space)* HyTime construct. It conveys that its descendents are placed in a coordinate system, in this case a system with a single axis named "street" which is measured in terms of virtual space units. It contains one *streetsched*, or HyTime *event (event schedule)*, which represents a single instance of the street coordinate system. The contents of a streetsched are *buildings*, conforming to the *event* HyTime construct. Each building construct refers to an image file. Its *exspec (extent specification)* attribute refers to the *streetnum* element that defines its street number. Each streetnum element contains the street number of a particular house. It conforms to the HyTime *extent* construct, which represents the position of one object along one axis of a coordinate system. The content of extents, and thus of streetnums, consists of a pair of numbers, the first stating where the object starts, and the second where it ends. If the second number is positive it represents how many units of measurement are occupied by that building starting with the first number. In Fiets, it is assumed each building occupies one street number, thus the second number is always 1.

Fiets Document

The code from the example Fiets document is shown in Figure 4. It specifies three addresses along the Herengracht, numbers 284, 308, and 334, and associates each with a JPEG file containing a picture of it.

```
<?ArcBase HyTime>

<!NOTATION HyTime PUBLIC
  "ISO/IEC 10744:1997//NOTATION AFDR ARCBASE
  Hypermedia/Time-based Structuring Language
 (HyTime)//EN">

<!ENTITY % HyTime PUBLIC
  "ISO/IEC 10744:1997//DTD AFDR Meta-DTD
  Hypermedia/Time-based Structuring Language
 (HyTime)//EN">

<!ATTLIST #NOTATION HyTime
  ArcDTD   CDATA   #FIXED "%HyTime"
  ArcDocF  NAME    #FIXED "HyDoc"
  ArcAuto  NMTOKEN #FIXED ArcAuto
  ArcOptSA NAMES   #FIXED "commatts sched"
  commatts CDATA   #FIXED "dafe dvllist ireftype"
  sched    CDATA   #FIXED "sched HyExSpec"      >

<!NOTATION jpeg SYSTEM>

<!ELEMENT fiets - O (streetfcs, streetnums)>
<!ATTLIST fiets HyTime NAME #FIXED "HyDoc">

<!ELEMENT streetfcs - O (streetsched)>
<!ATTLIST streetfcs
  HyTime NAME  #FIXED "fcs"
  axes   CDATA #FIXED "street virSpace">

<!ELEMENT streetsched - O (building*)>
<!ATTLIST streetsched
  HyTime NAME #FIXED "evsched">

<!ELEMENT building - O EMPTY>
<!ATTLIST building
  HyTime NAME    #FIXED     "event"
  exspec IDREF   #REQUIRED
  entity ENTITY  #REQUIRED       >

<!ELEMENT streetnums - O (streetnum*)>

<!ELEMENT streetnum - O (#PCDATA)>
<!ATTLIST streetnum
  HyTime NAME #FIXED     "extent"
  id     ID   #REQUIRED        >
```

Figure 3: Relevant Portion of the Fiets DTD

```
<!DOCTYPE fiets SYSTEM "fiets.dtd" [
 <!ENTITY h284 SYSTEM "h284.jpg" NDATA jpeg>
 <!ENTITY h308 SYSTEM "h308.jpg" NDATA jpeg>
 <!ENTITY h334 SYSTEM "h334.jpg" NDATA jpeg>
]>
<fiets>
 <streetfcs>
  <streetsched>
   <building entity=h308 exspec=numh308>
   <building entity=h334 exspec=numh334>
   <building entity=h284 exspec=numh284>
 <streetnums>
  <streetnum id=numh284>284 1</streetnum>
  <streetnum id=numh308>308 1</streetnum>
  <streetnum id=numh334>334 1</streetnum>
```

Figure 4: Example Fiets Document

Style Sheet

The code from the example style sheet is shown in Figure 5. It starts by generating a single *tuner* element, which specifies the window for displaying the pictures. Also generated are the required ancestor elements for the tuner, the *head* and *layout* elements. The style sheet then generates a *seq (sequence)* element, along with its required ancestor elements *body* and *par.* In SMIL, the contents of a seq are displayed one after another in time. The DSSSL code then finds all the building elements and then sorts them in order of street number. For each building in this sorted list a SMIL *img (image)* is generated, which refers to the image files and specifies that it be displayed in the window for a duration of 2 seconds.

Berlage SMIL DSSSL Library

The portions of code from the Berlage SMIL DSSSL library that are relevant to this example are shown in Figure 6. This code defines a function for generating a single SMIL img element, passing as parameters values to its attributes. This code defines the determination of the values of the HyTime properties used in the example.

Berlage HyTime DSSSL Library

The portions of code from the Berlage HyTime DSSSL library that are relevant to this example are shown in Figure 7. The property *firstq (first quantum)* is the starting point of an event along one axis with a coordinate system. The property *dim (dimension)* is the portion of an axis along which an object lies, usually represented with a first quantum along with a length. The *dimens (dimensions)* property conveys what portions of all the axes in a coordinate system an object lies along. These properties are referred to by the style sheet in this example to determine the street numbers of

```
<!DOCTYPE style-sheet
  PUBLIC "-//James Clark//DTD
  DSSSL Style Sheet//EN"
[ <!ENTITY je      SYSTEM "jade-extensions">
  <!ENTITY hytime SYSTEM "hytime.dsl"     >
  <!ENTITY smil   SYSTEM "smil.dsl"       > ]>
&je;
&smil;
&hytime;

(define (event-street-first-marker event)
  (node-property 'firstq
    (node-property 'dim
      (node-property 'dimens event) "street")))

(element building
  (img "map-tuner" "2.000s"
       (entity-system-id
         (attribute-string "entity"))))

(element fiets
  (make element
        gi:         "smil"
        attributes: (list
                      (list "lipsync" "false"))
    (make element gi: "head"
      (make element
            gi: "layout"
            attributes:
              (list
                (list "type" "text/smil-basic"))
        (make empty-element
              gi: "tuner"
              attributes:
                (list (list "id" "map-tuner")))))
    (make element gi: "body"
      (make element gi: "par"
        (make element gi: "seq"
           (process-node-list
             (sort-node-list
               event-street-first-marker
               <
               (select-elements
                 (descendants (current-node))
                 '("FIETS" "STREETFCS"
                   "STREETSCHED" "BUILDING")
))))))))
```

Figure 5: Example Style Sheet

```
(define (img loc dur href)
  (make empty-element
        gi: "img"
        attributes:
          (list (list "loc"  loc )
                (list "dur"  dur )
                (list "href" href))))
```

Figure 6: Relevant Portion of the Berlage SMIL DSSSL Library

```
(define (node-property propname node #!rest args)
  (if (string=? (symbol->string propname)
                (symbol->string 'firstq ))
      (car node)
  (if (string=? (symbol->string propname)
                (symbol->string 'dim    ))
      (list-ref node (axis-ndx node (car args)))
  (if (string=? (symbol->string propname)
                (symbol->string 'dimens ))
      (gen-dimens
        (string->number-list
          (data
            (element-with-id
              (attribute-string "exspec" node))))
        (string->name-list
          (attribute-string
            "axes" (parent (parent node)))))
  (literal "invalid property name")))))

(define (axis-ndx dimens axis-name)
  (if (< (length dimens) 1)
      (list "axis not found")
      (if (string=? (list-ref (car dimens) 2)
                    axis-name)
        (string->number "0")
        (+ (axis-ndx (cdr dimens) axis-name)
          1))))

(define (gen-dimens number-list name-list)
  (if (= (length number-list) 2)
      (list
        (list (list-ref number-list 0)
              (list-ref number-list 1)
              (car name-list)))
      (list
        (list (list-ref number-list 0)
              (list-ref number-list 1)
              (car name-list))
        (gen-dimens (list-tail number-list 2)
                    (list-tail name-list   2)))))
```

Figure 7: Relevant Portion of the Berlage HyTime DSSSL Library

the houses along the Herengracht that are represented in the document.

The way in which these properties are made accessible is by extending the definition of the standard DSSSL function *node-properties*. This function is recognized by any DSSSL engine and is used to return the values of certain SGML-related properties of SGML constructs. With the extension defined in the Berlage HyTime library, names of HyTime properties can now be passed to this function as well as those of SGML properties. This enables the reference of HyTime properties in a manner consistent with the properties of SGML and other property sets.

Not included in the code shown is the specification for the functions such as string->number-list, string->name-list, first-sub-string, rest-sub-string, min, remove-node, and sort-node-list. These functions have many uses for DSSSL code in general but are not part of the DSSSL standard. We have defined these in the Berlage HyTime DSSSL library for general use.

BROADER APPLICATION OF THESE STANDARDS, TOOLS AND LIBRARIES

The constructs discussed and the example presented in this paper touch on only a few of the representational facilities of HyTime. The HyTime constructs and properties used in the example refer only to determining a single coordinate of an object's measured placement in a coordinate system. One possible extension of this small example is representing the year of construction for each house. This would involve adding HyTime code for a "year" axis to the coordinate system and adding extent elements that specify these years for each house. With a small change to the current style sheet, the houses could be displayed in order of year of construction rather than by street number. Whereas the previous presentation gave the user the sense of walking along the street, this new presentation would give the user a sense of how the external architecture of Amsterdam houses changed over time.

HyTime hyperlinking could also be applied, for example, to associate each house with a historic figure who lived there. HyTime links could be used to structure information about the occupants, associating each with portraits made of them and with work accomplished by them such as paintings made or buildings designed. A DSSSL style sheet could then reference the properties represented by these HyTime constructs to generate SMIL code that enables, for example, a user to click on a house to display a portrait and descriptive text for one of its historically significant occupants. Such a style sheet would require extensions to the node-properties functions similar to those shown in Figure 7.

DSSSL has been mainly created for and applied to the transformation of text documents to page-based presentation. In other work we have discussed the use of DSSSL to address style issues that are particular to hypermedia [18][19]. The techniques discussed in this paper can be applied to implementing style sheets that address these issues following the principles described.

The Fiets environment or environments that use the same or a similar design can be used in electronic publishing to address common issues of importance to the electronic publishing community such as micropayment and the placement of restrictions on the presentation of individual media objects that adapt to their varying contexts within their final presentations [14]. As with hypermedia style sheet issues, The techniques discussed in this paper to addressing these electronic publishing issues.

SMIL is only one of the possible output presentation formats for Fiets and similar environments. Fiets can, by using appropriate style sheets, output HTML for hypertext presentation. It can also output formats for print such as PostScript. We have made style sheets that transform Fiets documents to the multimedia presentation format MHEG-5 [10], which provides fewer adaptive and synchronization facilities than SMIL, offers a more visually complex navigational interface than SMIL, and is designed for hypermedia presentations in minimal resource environments, such as settop boxes that work with broadcast television. We have also published a discussion on how DSSSL can be used to transform SMIL presentations to MHEG-5 presentations [17].

CONCLUSION

In this paper we introduced Fiets, an application that uses existing non-proprietary standards and tools to store and display presentation independent hypermedia data. Fiets demonstrates how the storage and presentation of distributed hypermedia can be readily implemented with the standards HyTime, DSSSL, and SMIL and the tools SP, Jade and GRiNS. Fiets's application of the standards and tools was illustrated with a simple example. The broader application beyond this example of these standards and tools was also discussed. This paper provides the reader with the information needed to implement similar digital library systems using the standards and tools described.

ACKNOWLEDGMENTS

The CMIFed and GRiNS environment was implemented by Sjoerd Mullender, Jack Jansen and Guido van Rossum. This work was funded in part by the MAGUS project, the Dutch Ministry of Economic Affairs, and the European Union ESPRIT Chameleon project.

REFERENCES

1. Bray, T., Paoli, J. and Sperberg-McQueen, C.M. (eds.). Extensible Markup Language (XML). W3C Recommendation, Janurary 1998.

2. 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 (WWW7), April 1998.

3. Clark, J. Jade — James' DSSSL Engine. http://www.jclark.com/jade/.

4. Clark, J. SP — An SGML System Conforming to International Standard ISO 8879 — Standard Generalized Markup Language. http://www.jclark.com/sp/.

5. DeRose, S. and Durand, D. Making Hypermedia Work: A User's Guide to HyTime. Kluwer Press, Boston. 1994.

6. Goldfarb, C. The SGML Handbook. Oxford University Press. 1991.

7. Hoschka, P (ed.). Synchronized Multimedia Integration Language. W3C Proposed Recommendation, April 1998.

8. International Standards Organization. Hypermedia/Time-based Structuring Language (HyTime). Second Edition. ISO/IEC IS 10744:1997, 1997.

9. International Standards Organization. Hypermedia/Time-based Structuring Language (HyTime) AFDR Meta-DTD. ISO/IEC IS 10744:1997//DTD AFDR Meta-DTD, 1997. http://www.hytime.org/materials/hi2mdhyt.sgm.

10. 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, 1997.

11. International Standards Organization. Document Style Semantics and Specification Language (DSSSL). ISO/IEC IS 10179:1996, 1996.

12. International Standards Organization. Standard Generalized Markup Language (SGML). ISO/IEC IS 8879:1985, 1985.

13. Rutledge, L., van Ossenbruggen, J., Hardman, L. and Bulterman, D. A Framework for Generating Adaptable Hypermedia Documents, in Proc. ACM Multimedia 97, November 1997.

14. Rutledge, L., van Ossenbruggen, J., Hardman, L. and Bulterman, D. Addressing Publishing Issues with Hypermedia Distributed on the Web, in Proc. ICCC/IFIP Electronic Publishing 98, April 1998.

15. Rutledge, L., van Ossenbruggen, J., Hardman, L., Bulterman, D., and Eliëns, A. Use of Standards for Hypermedia Generic Structure and Presentation Specifications, in Proc. ICCC/IFIP Electronic Publishing 97, April 1997.

16. Staflin, L. A GNU Emacs mode for SGML files. 1996. http://www.lysator.liu.se/projects/about_psgml.html.

17. Ten Kate, W., Bulterman, D.C.A., Deunhouwer, P., Hardman, L. and Rutledge, L. Presenting Multmedia on the Web and in TV Broadcast, in Proc. 3rd European Conference on Multimedia Applications, Services and Techniques (ECMAST 98), May 1998. (submitted)

18. Van Ossenbruggen, J., Hardman, L. Rutledge, L. and Eliëns, A. Style Sheet Support for Hypermedia Documents, in Proc. Hypertext 97, April 1997.

19. Van Ossenbruggen, J., Hardman, L. Rutledge, L. and Eliëns, A. Extending Style Sheet Languages to Hypermedia, in Proc. Hypertext 98, April 1998. (submitted)

20. Van Rossum, G., Jansen, J., Mullender, K.S. and Bulterman, D.C.A. CMIFed: A Presentation Environment for Portable Hypermedia Documents, in Proc. ACM Multimedia 93, August 1993.

21. World Wide Web Consortium. Synchronized Multimedia Integration Language Document Type Definition. January 1998. http://http://dejavu.cs.vu.nl/~symm/validator/SMIL10.dtd.