HDML: an Object-Oriented Hyper-document Descripting Meta Language for E-education

Yolaine Bourda, Bich-Liên Doan, Henri Delebecque

HDML: an Object-Oriented Hyper-document Descripting Meta Language for E-education

Henri Delebecque, Bich-Liên Doan, Yolaine Bourda
Supélec
3 Rue Joliot-Curie
F 91190 Gif sur Yvette FRANCE
Henri.Delebecque@supelec.fr, Bich-lien.Doan@supelec.fr, Yolaine.Bourda@supelec.fr
http://www.hdml.org

Abstract: The media on which authors should publish pedagogical documents become every day more numerous, complexifying the work of either content authors, graphics designer or webmasters.

We propose an XML application for content authors, to help them structure their hypertext documents in factorizing their common structure in a class, and to free them from all the other aspects. They design both a structural model (a class), and a presentation model (using HTML, WML,…). Using these two models, our HDML engine automatically builds the projection function able to transform any actual document into (for example) well-formed web pages.

HDML also offers inheritance capabilities among classes, which allows an editor responsible for a whole document set to define its global structure, and to adapt it according to the different document sections needs.

1. CONTEXT

1.1 Hyper Document Structuring Language

XML can be used for the definition of a hypertext document structuring language like HDSL (Delebecque 2002), with multiple benefits:

- ability to add semantics to structural elements,
- factorization of the structure common to a document set in a structural model (the class),
- clear separation between content and presentation, using a presentation model for each presentation medium,
- ability to maintain a coherent look-and-feel all through the document set, on all media

1.2 Why Object Orientation in HDML

The inheritance feature object-oriented languages define increases the power of a document class definition in allowing:

- factorization between structural models,
- customization of a general (e.g. book-wide) structure into a more precise (e.g. author-specific) one,
- specification of the operations applied to a specific document or element.

1.3 HDML General definition

HDML stands for "Hyperdocument Description Meta-Language". HDML's goal is to give content authors the ability to focus on content and ignore presentation, while using their favorite editing tools. HDML is clearly a meta-language, since it allows the class designer to define a markup language that will be used for writing documents in the content authors domain. This language definition can be considered an instantiation of HDML in this specific language, that we will call the content authoring language.

2. PRINCIPLES

HDML can be used by authors in three different ways, while defining three kinds of documents, which are:

- the class (the structural model),

- the presentation model,
- the concrete document (a class instance).

The class designer, who is also the whole pedagogical document editor, defines the structural model common to all the site documents. The graphic designer, or the ergonomist, designs the look-and-feel used to display these structural elements in a specific presentation medium in a presentation model. Lastly, the content author instantiates the class in concrete documents.

The HDML engine works in two stages, using the three different kinds of documents described earlier. To allow for the separation of semantics and presentation, the "Holy Grail" of XML, we propose a global lightweight framework working with pairs of documents. For every subset of the global pedagogical document (book, chapter, section…) which may or may not have its own look-and-feel, the document editor supplies a class using HDML, and the graphic designer a presentation model for each presentation medium.

Our HDML engine takes all these pairs of models and then automatically generates as many XSLT style-sheets as needed to produce "displayable" forms of the concrete HDML documents in the various presentation media forms. Thus, every XSLT style-sheet can be considered a projection function from the class to the presentation model. Subsequently this projection function is applied to the concrete documents that conform to the class structure, producing the published documents. These later documents can of course include additional items (e.g. background, logo…) provided by the presentation model.

3. INHERITANCE IN HDML

A document family definition can itself be derived from another wider family. This is useful for example if the former is a chapter of the book described by the later, or the web pages of a chapter in the web pages of a whole course. XML Schema provides a simple but useful inheritance feature among types (Davidson et al. 1999). HDML goes much further.

3.1 Structural Inheritance

A class can inherit its structure from another class, which in turn can inherit from another, as many times as required. In this case, we call the class inheriting a subclass, and the set of classes from which the subclass inherits the superclasses, as in object-oriented languages. The subclass can add new elements, replace (overload) or delete some elements one of the superclasses defines.

This deletion capability can have multiple practical applications. First, it will simplify the authoring task, by removing useless structural parts. The content author can thus focus on the elements relevant for him. Lastly, it increases the global document coherency by insuring its editor that critical element values (such as the version, date…) are hidden from content authors, and thus not modified in concrete documents. At the same time, HDML proposes a locking feature to the class designer, to prevent the deletion of mandatory elements in a subclass.

3.2 Behavioral Inheritance

The concept of method, familiar in object-oriented languages, could seem strange in HDML, due to its lack of programming features. Let us define it by analogy: we know some obvious operations that can be applied to documents, for example, the document initialization, or projection into a presentation medium (using an adapted presentation model).

The class designer can define a special element behavior with a method. Using this feature, he can initialize or project an element in a different manner than another, even of the same kind. This feature could be used, for example, to display the exercise answer only after the third student guess. More generally, the author can control the published information, even at the finest level, with minimal cost since this kind of information can be easily factorized in the class.

The ability to define an element method is also useful for splitting the HDML engine projection output stream into separate documents. The HDML engine is intended to take one content document, and to transform it into a "displayable" document. In many cases, this one-to-one association is not suited to the presentation medium, since the granularity of content authoring is different from the granularity of the

presentation. For example, an HTML page is more or less screen-length sized, whereas the content document is more likely divided into structural entities like chapters or sections.

A document method can be defined in two different and complementary ways. The first way is that the author can overload the default behavior either in the class, or in the concrete document, for example, by specifying an instance-specific presentation model. A second way of defining a method is implicit, using inheritance. Classes inheriting their instance structure from superclasses will also inherit their instance behavior, as in object-oriented languages.

4. RELATED WORK

4.1 Content Management Tools

HDML is not designed to compete with commercial all-in-one content management products, like Rhythmyx, Vignette or others, even if they come from the academic world like WebML (Ceri et al. 1999), if these tools are based on proprietary formats or tools.

These products are more or less devoted to the construction of commercial web sites. Their content is often built from very dynamic information fetched from databases, and linked by complex relationships. For example, WebML uses a structural model that describes these relationships using an E/R model or an ODMG object-oriented model. There is also, as in HDML, a presentation model that defines the appearance of pages independently of the output devices and the rendering language. But WebML uses an abstract XML syntax for its definition. This option, like many others, suggests that WebML's application domain is more oriented towards e-commerce, since it requires heavy post-processing stages that HDML avoids.

HDML is mainly an open and lightweight tool, designed to keep users in their favorite environment. A major effort has been made to push its versatility at the syntax and operational levels and also to improve its importing and exporting capabilities.

4.2 Document Structuring Languages

Other projects have defined document structuring languages, as XML applications or not, such as DocBook. Even if they are very good for relatively simple document authoring, they lack the advanced features HDML offers, such as inheritance and content authoring language definition. Such capabilities are of great interest in reducing the global document management cost, especially when many presentation media are involved (like paper, slide or computer-based-support one in a pedagogical context).

4.3 XSLT

The work the HDML engine does can clearly (and perhaps better) be done by a XSLT programmer, eliminating the need for HDML. We will answer that XSLT is a powerful but not intuitive language for a non-professionals, which is the case for the majority of content authors. This implies a fourth kind of knowledge the authoring team should have, very deeply involved in the design of documents published, since this XLST author must provide the link between the structural and presentation models.

5. BIBLIOGRAPHY

Ceri S., Fraternali P., Paraboschi S. *Data-Driven One-To-One Web Site Generation for Data-Intensive Applications*. Proc. VLDB'99 September 1999. http://webml.elet.polimi.it/webml/

Davidson A., Fuchs M.,Hedin M., Jain M., Koistinen J., Lloyd C., Maloney M., Schwarzhof K. *Schema for Object-Oriented XML 2.0*. http://www.w3.org/TR/NOTE-SOX

Delebecque H. *HDSL: An e-education Hypertext Document Structuring Language*. ED-MEDIA 2002 Denver CO USA

Walsh N., Muellner L. *DocBook: The Definitive Guide*. http://www.docbook.org/