

Temporal and version extensions to XML and the W3C Document Object Model



8th International WWW Conference
Toronto, Ontario, Canada
11-14 May 1999

Tony Allan (tony@apms.com.au)
Gayle Allan (gayle@apms.com.au)

Allan Project Management Services

Introduction

- Asynchronous collaborative authoring is not well supported by current systems.
- One way to overcome these limitations is to:
 - Add XML tag attributes to support temporal (history) and user information.
 - Extend the Document Object Model (DOM) interface by adding context information.
 - Utilise client/server editors.
- A prototype has been created that allows multiple authors to create and edit documents using a very simple web form based editor.

Issues with Existing Editors

- Each version is treated as a separate document.
- Changes made using revision marking facilities are lost once the revisions are accepted.
- A document lifecycle consists of several drafts and modifications often by a number of people over a period of time. When published, the bulk of useful information collected during a documents production is lost.
- Any embedded comments and notes are removed and lost at time of publication.
- Separate documents are often maintained.
- Delays in gaining write access for local shared files and hard to manage duplication for remote access (i.e. email).

Objectives

- To allow a new author or an author who has been absent from a team for a period of time to be able to quickly come up to speed with why and how the document evolved to its current form.
- To allow discussion and annotation by readers to be attached to the desired part of the document and retained in sequence.
- Note:
 - A document is anything that can be expressed as XML, such as text, diagrams or even a spreadsheet.

Goals

- Retain a complete history of changes from document creation including when the change was made and by whom.
- Retain discussion and other annotations.
- Allow multiple concurrent updates.
- To allow interesting meta-visualisation of documents, eg. which authors contributed to what parts of the document and how the document has evolved over time (ie finding the various locus of change).

Issues

- XML document conforms to its DTD at any single point in time but the overall document (with all history) does not.
- Multiple versions of a document in a single XML file, with ongoing changes to each version, is difficult to support.
- Full functionality requires new WYSIWYG editors operating in a client/server environment.
- The social issue of privacy which can be partially overcome by restricting access to some information.

Traditional CSCW categorisation

	Same time	Different times
Same place	Face-to-face (classrooms, meeting rooms)	<i>Asynchronous</i> interaction (project scheduling, coordination tools)
Different places	<i>Synchronous</i> distributed (shared editors, video windows)	<i>Asynchronous</i> distributed (email, bulletin boards, conferences)

- Document creation tends to be asynchronous.

Scenario - Web-based Journal

- Objectives
 - Increase the number of people who provide content or review information in the journal.
 - Eliminate the need for HTML expertise.
 - A managed, periodic publication process, with the ability to correct or add to previous issues of the journal after publication (the proposal doesn't fully support this).
 - Complete separation of content and design.
 - Allow direct, controlled discussion and feedback by user group members.

Scenario - Corporate document

- Objectives
 - Allow a changing group of authors to create and maintain a document over a period of months or years.
 - Allow a new author to understand the reasoning behind the current document content.
 - A managed, periodic publication process.
 - Complete separation of content and design.
 - Allow direct, controlled discussion and feedback by the authors, reviewers and readers.

Solution

- Demonstration software built as a simple Java HTTP server and application.
- User interface is an HTML form-based editor and HTML document viewer supporting multiple user access.
- DTD is currently hard-coded.
- Granularity of change is a DOM node although attributes would only be associated with an XML tag when the data is saved.

Extensions to DOM

- The design is based on the W3C Document Object Model (DOM) .
- A context object has been added which:
 - contains a version and a date.
 - is used with methods that retrieve nodes from the document.
- The context object is passed to the standard navigation objects which then return the appropriate node or nodes.
- New methods include:
 - `inFocus(Context c)` is a method that returns true if the Context date is between the start and end dates of the node.
 - `getNodeID()` and the Document method `getNode(String id)` allow access to an internal node id attribute which is used to simplify processing of HTML forms.

Conclusions & Further Work

- DOM represents a robust interface and the extensions needed are relatively minor.
- The web is suitable for read-only access and relatively simple editing tasks. Good URL design is an important part of a web interface.
- More complex editing will more easily be performed with a user interface modelled after traditional editors.
- Further work:
 - Performance and capacity issues with very large documents or document collections.
 - Design of intuitive web form based user interfaces.
- For further information, email the authors.