Chapter XVI
Document Versioning and XML in Digital Libraries

M. Mercedes Martínez-González
Universidad de Valladolid, Spain

INTRODUCTION

Digital libraries are systems that contain organized collections of objects, serving in their most basic functions as a mirror of the traditional library that contains paper documents. Most of the information contained in the collections of a digital library consists of documents, which can evolve with time. That is, a document can be modified to obtain a new document, and digital library users may want access to any of those versions. This introduces in digital libraries the problem of versioning, a problem that is also of interest for the hypertext community and the Semantic Web community. Some domains in which document evolution is a very important issue are the legislative domain (Arnold-Moore, 1997; Martínez González, de la Fuente, Derniame & Pedrero, 2003a; Vitali, 1999), the management of errata made to scientific articles (Poworotznek, 2003), software construction (Conradi & Westfechtel, 1998), and collaborative e-learning (Brooks, Cooke & Vassileva, 2003).

In the legislative domain, rules suffer amendments that result in new versions of the amended rules. Access to all versions of a document is an important facility for their users; for example, to understand a tribunal sentence it is necessary to get access to the text of involved rules, as they were valid at the moment the sentence was made. Errata to scientific articles are somewhat similar. The errata are posterior to the original article and they are published together with the reference to the part of the article to be changed by the modification. In software construction, different versions of program files are available at the same time, and the composition of software has to assemble adequate versions in order to obtain the correct version of the software. In e-learning frameworks, the problem comes from the updates

made to content objects, or the reordering of these contents.

In recent years, the spread of XML as the metalanguage for document modelling has been accompanied by a strong interest in XML document versioning. The interesting issue is that XML documents are no longer considered as atomic items that can be substituted or not, but composed of document nodes (elements) that can themselves be versioned. Besides, there have been several initiatives that propose using XML as the ideal format to represent metadata related with changes.

Next, we revise the issues related with document versioning, the main approaches proposed and the issues that each approach favours. Issues related with XML will receive special attention in this updated chapter[1]. Versioning a document impacts not only the document itself but also other items, such as references from and to the versioned document, or the indexes created for information retrieval operations.

BACKGROUND

As for the issues of interest related to document versions, we distinguish seven categories:

1. What can be versioned?

This question can be considered from two perspectives. The first perspective considers objects stored in the system as atomic units of information, which cannot suffer partial changes. This is the typical situation in the Web and hypertext environments. Hypertext nodes (documents, files, others) can change (be substituted, deleted, inserted), and the hypertext structure can also change (objects may vary their location, some of them may disappear, others may change their references to other objects), but each document is considered an atomic item which is not subdivided in other objects: changes always concern the whole document. The evolution considered in the second perspective is the one of the documents used by digital library users --these documents may or may not match unidirectionally any of the objects stored in the digital library (Arms, 1997)—and with XML documents. Changes in this case can be related with any component of a document: its content, part of it (e.g., some nodes in XML documents), the internal structure of documents, or references (citations within documents, that are part of a document).

2. Detecting changes

Sometimes it is necessary to recognise two documents as versions of the same work, or to find the changes that have to be applied to a document version to obtain another one. There are two possible ways to do this: extracting references from document content (Thistlewaite, 1997; Martínez, de la Fuente, Derniame & Pedrero, 2003a), or comparing versions (Chawathe, Rajaraman, García-Molina & Widow, 1996; Lim & Ng, 2001; Cobena, Abiteboul & Marian, 2002).

3. Representing changes

The information about versions and the changes between them has to be stored somehow in the system. This is dealt with in the version control model used, or in the data model, as happens with XML documents. Besides, metadata that describe the items can also be used to represent the versioning information. In summary, the main possibilities are:

- To store the versions caused by a change, and/or the corresponding differences (deltas). This is the classical approach in version management, and corresponds to solution 1 of versioning management approaches presented in the next section.
- To represent changes as annotations (attributes) to the versioned items. In this case