Developing a Framework for Building Open Distance Learning Websites in the Literature and Culture Domain

Alexander Schatten, A Min Tjoa
Insitute for Software Technology, Vienna University of Technology
(aschatt,tjoa)@ifs.tuwien.ac.at

Abstract
The abstract text, ...

1. Introduction

The German Language and Literature Department of the Paris-Lodron University Salzburg detected an increasing interest in Austrian Literature round 1900 and Exile Literature. First ideas to build this site with standard tools were rejected, because of the complexity and quantity of material to be presented. Simply writing the lectures would have produced HTML documents in coarse numbers, which would have been difficult to manage, especially in respect to design, adding new material and production of different publication formats (e.g. PDF, e-book).

Furthermore reuse of content would have been hardly possible. Hence the German Literature and Language Scientists started a cooperation with the Insitute of Software Technology to build a more general framework. This was also an explicite desire of Ministry of Science, that founded the projects.

The aim of the paper is to describe technical development on the basis of the project-workflow, to show a realistic scenario of the formation of such a content management and publication framework.

2. Workflow

2.1. Introduction

In the following section the workflow of the first project, which was also the prototyp for subsequent projects, is described in chronological order. The problem was, that we had not much time for technical plannings and programming in advance as the German language and literature scientists needed to enter data into the system. So in a three step approach we first implemented a prototyp of one part of the first project, then we finished the prototyp for the first project and hence in the end of the first and the initial phase of the subsequent second project we consolidated and re-designed parts of the implementation of the database, publication and user interface tools.

2.2. Initial Plans

A more close look to the desired Open Distance Learning (ODL) Courses offered the demand to:

- Publish content (lectures and biographical content) in the amount of about 500 printed pages
- Manage hundreds of images, photos and other multimedia content (literature references, original text, music, ...)
- Publish to different output formats (HTML, PDF, eventually Open E-Book)
- Be flexible in reuse of the content
- Be flexible in design
- Be open for further extensions (interactive elements for example)

Hence in cooperation with the Ministry of Science the original plan to provide information on the Internet was exteded to first of all build a more general framework for managing the information and publication in a flexible way.

2.3. Design

Moreover design was an important issue: Not only should a designer cooperate with the technical team, also design should be consistent within the complete project. The technical framework is the basis to allow easy modification and adaption to the design of all elements of the ODL information pool.

2.4. Technical Prototype

The first step was building a technical prototype considering the „easier“ part of the project: the biographical information. We implemented the basic framework for this part of the project to test whether all parts of the process would work as planned:

- Input of the data (should be performed by German literature scientists)
- Stable data storage
- Data Exchange and Metadata
- Publication in multiple formats including Design Stylesheets

2.5. Design Prototype

Parallel to the development of the technical prototyp the designer implemented a design prototyp of all parts of the project. This was not only important by esthetical but also by technical reasons: It was necessary to find a (small) set of design elements to build the design templates for automatic generation of the website. No manual building or modification of pages should be done (especially for very few special informations).

2.6. Coordination of Concepts

Having successfully implemented these prototypes we again coordinated our conceptions, but it became clear, that all steps of the workflow do indeed work as desired, also the most critical step: the data input procedure. The next steps were the design and implementation of the more complex data structure (lecture information).

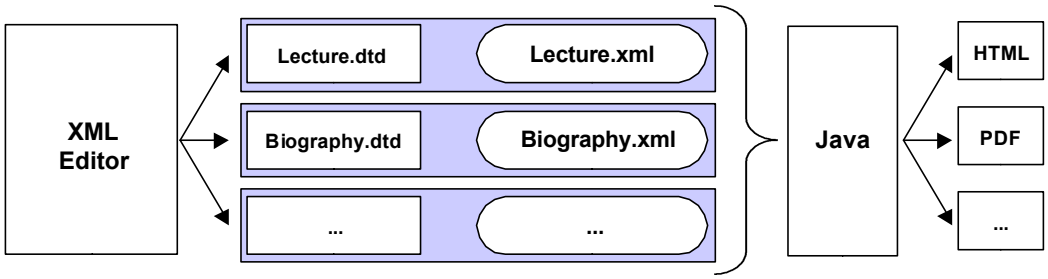

Figure 1 *Process of Data Storage in XML, Editing and Publication*

2.7. Alpha/Beta Test, Final Release

The alpha test of the first project was performed internal, on password protected web-pages, the beta test was public. When the final release[1] of the first project started, the following project was in its initial phase and we started the re-implementation of the database and the redesign of the User Interface besides other enhancements like adding interactive elements to the ODL website.

3. Technical Details

3.1. XML

It was clear from the beginning of the project, that XML (Bray 1996, Harold 1999) should play an important role as data format. As mentioned above a highly generic approach in collecting and storing the data was necessary. As the data strucutre is rather complex (especially when looking at „lectures“), and flexible publication from XML is possible, we decided to use XML as data format.

XML has the advantage of beeing an open standard with many parsers available. As non-binary format including meta-information with the data, the constraint of persistent data storage is fulfilled.

The first idea was to directly use XML when entering the data. In fact at the start of the project we could not find XML editors that are good enough (and affordable) that non-computer experts could easily use them for entering the lecture and biographical data. (This idea is illustrated in Fig. 1.)

3.2. Relational Database Management System

Keeping the data consistent (including the multimedia entities and references between content elements) would have been a complicated task with the existing XML editors (though recent XML enhancements like schema support could support these verification). For this and other obvious reasons, we decided to build the data structure using an RDBMS desktop system (to allow rapid prototyping), e.g. using referential integrity check of the database to guarantee the consistency of the data to a certain degree. Entering the data is done by a GUI programmed with the desktop database system at the first project.

[1] http://www.sbg.ac.at/lwm

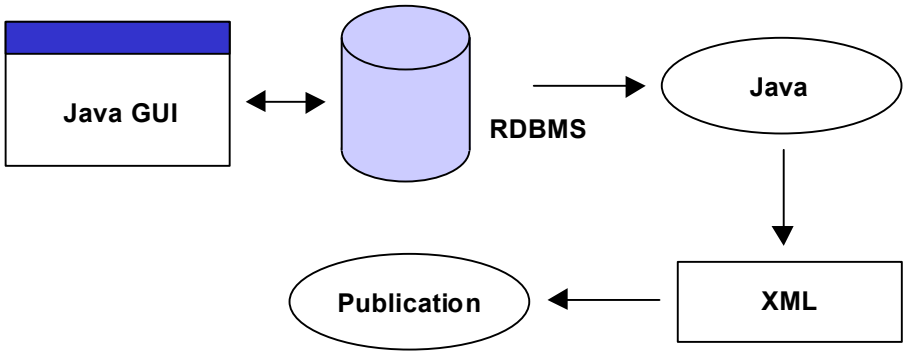

Figure 2 *Database-driven Implementation with User Interface that accesses the RDBMS system via network. The XML files are generated with Java Applications. The Publication follows the illustration in Fig.1.*

In the second project (Exile Literature) we moved the data to a database server (Interbase) and implemented the GUI in Java for easy portability and the ability to create a multiuser system.

The XML data is generated automatically out of the RDBM system. The advantage of this approach is that the data stored in the database is non-redundant and consistent, the data in the XML file is including meta-information, publication oriented and portable. Generating the XML files automatically furthermore guarantees that the XML files are well-formed as well as valid (compared with the appropriate document type definition). The basic procedure is as illustrated in Figure 1, with the modification, that the XML files are not written with an XML Editor but generated out of the database.

3.3. Metainformation

The XML files are self-desciptive by nature. Furthermore we use RDF (ressource description framework) metadata in the biographic HTML files following the Dublin Core specification (Dublin Core 1997).

The use of metadata may help to integrate the project into other frameworks like bibliographic systems or in search engines or agent environments. Generally spoken our generic approach allows adaption of arbitrary metadata systems if needed.

3.4. Portable and Platform Independent Approach

Our framework is not only highly generic, but also platform independent, portable and open, especially considering the last modifications:

- As RDBMS system we use the Open Source database Interbase, which is available for (at least) Win32, Linux and Solaris, currently running on Linux Server
- XML Files are „simple text“ files, hence portable ans platform independent by nature. Reuse of data in other context is easy using these XML files.
- As „scripting“ and programming language for GUI, conversion and publication purpose we use Java, JDBC database access and JAXP XML parser (Sun 1999); the system is tested on Win32 and Linux, but should run on any Java 2 platform
- As Publication formats HTML (with litte Javascript) and PDF are used. These formats are usable on all relevant operating systems.

3.5. Publication

Publication to HTML, PDF and Open E-Book is performed using the XML files and Java applications. Using the XML files is easier than directly publishing out of the database, as the XML files are already publication oriented, meaning, that the information is already prepared for publication, in opposite to the database system where non-redundancy has priority.

At the moment HTML files and PDF are statically generated and published using a standard webserver. In future it could be possible using XML directly, especially when browsers allow stable rendering of XML using stylesheets (e.g. XSL stylesheets).

The open e-book format is also a XML format, hence publication to Open E-Book is basically only a transformation from one DTD to another.

Our object-oriented framework allows producing new publication formats by simply adding new driver classes for the desired output.

3.6. Design Templates

Necessary for HTML publication and flexible design was the definition of HTML templates. We defined a „variable“ syntax allowing to modify the test HTML pages produced by the designer to design templates by adding these „variables“ into the HTML document.

On publication, the Java application reads the HTML templates and replaces the variables with the appropriate values (e.g. biographical content...). This approach allows to change the design of the complete project by simply exchanging a set of desing templates and regeneration of the static HTML files.

4. Documentation and Communication

4.2. Communication

As the teams (German language and literature scientists, programmer, designer) were located on different places in Austria communication was a very important factor for success of the project. Besides the „conventional“ communication channels like meetings (which were limited because of dislocation) and telephone, we extensively used electronic comminication channels: email, web-based documentation and groupware (BSCW), and Internet Newsgroups (with authentification) to enhance technical and content oriented discussion and as document archive.

4.2. Team Documentation

Each team maintained a web-based documentation of their part of the project, containing the specific information of the team.
All results (Java Programs, database structure, database, publication HTML and PDFs, ...) were first published on the internal documentation pages to inform all project members and also the Ministry of Science about concepts, the implementation progress and problems occuring.

4.3. Project Monitoring

All these asynchronous communication media (especially webpages and newsgroup) offered an easy possibility for project management to monitor the project status and get

an easy overview. Moreover we maintained a technical „todo“ list to show open issues and finished tasks (with date). Hence problems in project progress could be detected early enough for taking correctional measures.

5. References

- Bray, Tim and Sperberg-McQueen, C.M. (1996), *Initial XML draft*, http://www.w3.org/TR/WD-xml-961114.html
- BSCW: Basic Support for Collaborative work: http://bscw.gmd.de
- Dublin Core (1997), *Dublin Core Metadata Initiative*, http://purl.org/dc
- Ressource Description Framework, W3C Initiative: http://www.w3c.org/rdf
- Harold, Elliotte Rusty, *XML Bible*, IDG Books (1999)
- Lassila (1997), *Introduction to RDF Metadata, W3C Note* (1997), http://www.w3.org/TR/NOTE-rdf-simple-intro-971113.html
- Leventhal, Michael and Lewis, David and Fuchs, Matthew, *Designing XML Internet Applications*, Prentice Hall PTR, Upper Saddle River (1998)
- Sun Microsystems (1999), *JavaTM Technology, XML: A Perfect Match*, http://java.sun.com/xml/