REFACTORING THE APPLICATION INFRASTRUCTURE FOR BUILDING OPEN-DISTANCE –LEARNING WEBSITES IN THE GERMAN LITERATURE AND LANGUAGE DOMAIN

ALEXANDER SCHATTEN, MARIAN SCHEDENIG, A MIN TJOA

Institute for Software Technology and Interactive Systems, Vienna University of Technology, Favoritenstr. 9-11/188, 1040 Vienna, Austria
E-mail: {schatten, schedenig, tjoa}@ifs.tuwien.ac.at

Growing interest in german literature initiated two succeeding projects to provide content about "Austrian literature round 1900" and "Austrian Exile-Literature since 1933" as well as the technical infrastructure for content management and publication. This infrastructure is platform independent and based upon open standards like XML and mainly on open source libraries and tools. Publication is done to multiple formats like HTML and PDF from the same sources. From the first project to the second project it turned out, that standards and tools were subjected to significant changes, as well as experiences from the first project led to new concepts in the second project. Hence a complete refactoring of the application-code from the first project was performed. The experiences of this multi-step process is the main focus of this article.

1 Open Distance Learning - Literature in Internet

In the past years a steadily growing interest in german literature was experienced. Hence the department of german literature and language science of the University Salzburg initiated two open-distance learning projects in cooperation with the institute for software technology and interactive systems of the Vienna university of technology. This paper describes the technical aspects of these projects, mainly focused to the refactoring steps between the two succeeding projects.

The first project, „Literatur in der Wiener Moderne" (Literature round 1900) was started in Spring 1999 and finished in summer 2000 [8], with the (technical) goal to create a content management and publication system build with open standards beeing highly flexible in the publication process. The second project „Exilliteratur" (Exile-Literature) was started immediately after the first project and ended in spring 2002.

The first project now has about 600 registered users and more then 100 exams made by students, with a focus not only to students in Austria, but also to eastern countries like Russia, Croatia, ... The information provided by this project turned out to be a valuable and accepted source for students in different countries interested in this specific area of literature. The second project will start the „public phase" in the next month, so no practical experiences are available yet.

2 Literature Projects

2.1 Basic Concepts

The content is based on statical created html pages for main navigational purpose or few special features. The main content is managed in a content management sytem published automatically using templating mechanisms.

The automatically generated information is structured mainly into lectures, articles and multimedia elements (images, original texts, video, urls, ...). Where lectures are bigger in size then articles and contain the „main information“ to learn from. Articles are additional information; for example articles about authors, or specific „side-topics“. From the first to the second project both categories have been extended. While „articles“ meant information about authors in the first project, different types of articles (extensible) are available in the second project. Also different types of lectures are available in the second project. This data is provided twofold: for easier online reading in html „chunks“ and for printing in portable document format (PDF).

Besides the automatically generated information there are additional (statical created) sections like special prepared topics in „museums“ or a „walk“ through famous literature coffe houses in Vienna.

Keywords are registered during the entering of the information, not by automatical indexing mechanisms. As a consequence, the keywords are of very high quality and provide a valuable additional access schema for users.

In the second project „interactive elements“ as well as new multimedia types (video, audio) have been added. The interactive elements allow students-interaction with the system by answering different types of questions. A deeper understanding of the lectures is expected by the use of this new feature.

2.2 Cooperation

The project successful cooperation was a a demanding task by many reasons. Not only the communication between different „scientific cultures“ namely information technology, german literature and language science and design needed time and professional engagement on all sides. Moreover the teams were dislocated: Literature and Language Scientists are located mainly at University of Salzburg, Graz, but also in Vienna, information technology and design is located in Vienna.

Good project management, and the efforts on all sides to communicate even difficult tasks in a way, that all project members are able to understand the consequences, as well as using all possible communication facilites leaded to successes in both projects.

2.3 Focus of Projects

The Open Distance Learning Project „Literature round 1900“ was the initial project and build the foundation of cooperation and concepts as described above. The Online-Project “Österreichische Exil-Literatur seit 1933" (Austrian Exile-Literature since 1933) is dealing with emigration and exile in the 20th century. The projects aim to present qualified material anbout the topics to students and to the interested public in general. The educational material of this project is a foundation for approaching this topic in higher schools as well as in universities and education for adults.

Especially for the open-distance-learning, interactive elements are designed to provide the possibility of a deeper interaction for the students with the material. Working with these interactive pages are also a part of the evaluation process in examinations.

Part of the open-distance-learning project is also a cooperation with other universities for students to make exams about the exile-literature topic (e.g. universities in Innsbruck, Debrecen, Vechta, Leiden, St. Petersburg, Rijeka, VU Amsterdam).

2.4 References

The literature projects can be found under: http://www.literaturepochen.at (which is the main site). It has to be notified, that the main part of the information is in german.

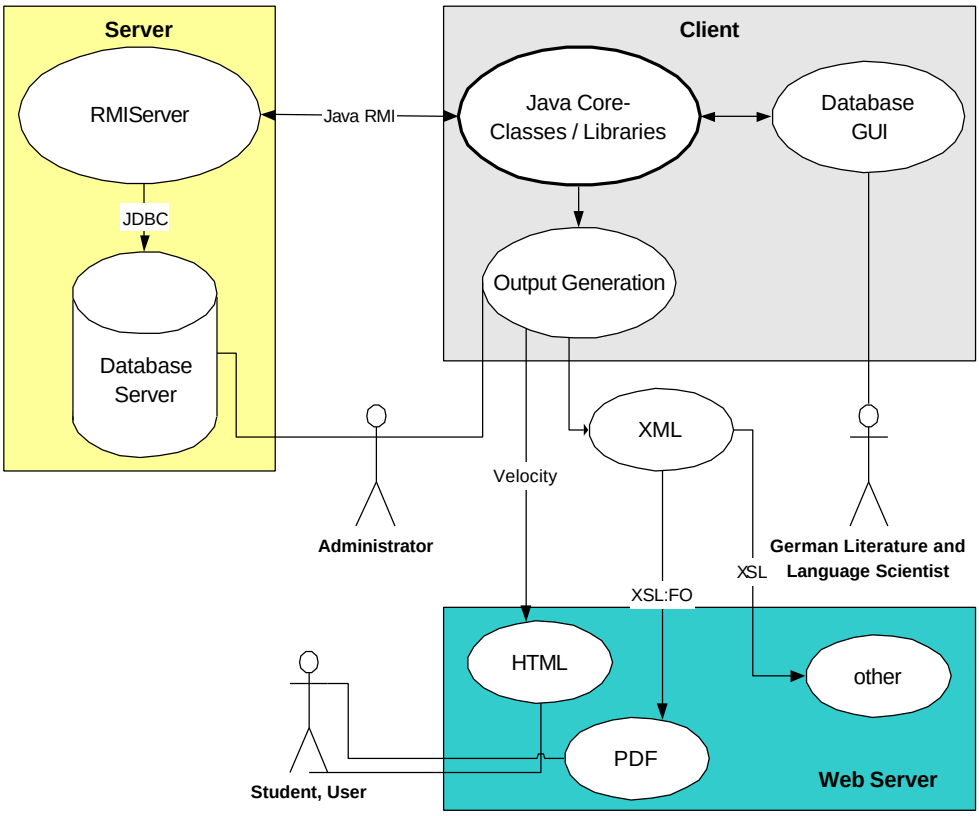

Fig 1. Conceptual Overview over the Content Management System

3 Used Technology and Experiences

3.1 Relational Database Management System (RBDMS)

Data is stored in RDBM systems. This turned out to be a useful approach to ensure the flexible access to the information and the integrity of the complex data structure. Moreover the modular approach allowed the easy move from a desktop database system in the first project to a RDBM server in the second project. We decided to use an open source database server (Interbase).

3.2 Java

Platform independent use of all parts of the application was a goal in both projects. The use of the Java programming language turned out to be a good decision by many reasons: First of all it was possible to run the applications and libraries on Windows clients as well as on Linux server systems. Additionally a huge pool of (mainly open source) tools and libraries are available (e.g. the Apache pool) which facilitated the implementation of several parts of the application. Finally Java allows good object oriented design and development of stable and very performant applications that run on different target platforms.

Connection between the different system parts (database, GUI-client server communication) was performed using standard Java protocols like Java Database Connectivity (JDBC) and Remote Method Invocation (RMI).

The data is published out of the database, partly using XML as mentioned later. These steps are performed using Apache java-xml parser, xslt and xsl:fo libraries and template engines [5].

3.3 Client / Server Architecture

In the first project the user interface was part of the desktop database, which turned out to be fine for rapid-prototyping, but a hard to maintain implementation for productive use. Consequently also the user interface was rebuild in the Java language, which allowed the re-use of libraries needed for different parts of the project. Moreover also the user interface is platform independent running on Windows, Unix and MacOS systems.

3.4 XML

The use of XML [4] is an essential part since the beginning of the first project. There are multiple reasons, why choosing XML proved to be a good decision. First of all, the produced content is extremly valuable also in future, honestly seen, eventually more valuable then the technological infrastructure. By transforming the complete information into XML a future use also in completely different systems will be possible, hence the survival of the content should be guaranteed.

Moreover using XML as intermidiate format gives us the flexibility for publishing different output formats as described below [7].

3.5 Template Engines

The production of html is done using template engines. This allows the easy integration of sample html documents produced by the designer into the production system. Moreover, exchange or modifications of the design is possible also very late or even at the end of the production cycle.

3.6 Output Format

As mentioned already the two main output formats (for production use) are html (for on-line learning) and portable document format (PDF) for printing the information.

In research versions even other output formats were tested. The use of XML as intermediate format allows flexible and easy production for other systems, like open e-book or version for other mobile clients like PDAs. Test-productions for the rocket e-book were performed using XSL transformations and basically worked good. However, e-book formats were not yet integrated into the production version, as no significant impact is seen in the market (and even worse, different formats are available) and the students could be more confused by this option.

The „raw xml“ format could be offered directly or in (xsl) modified version in future editions for data exchange purpose or web services (library systems...), if necessary or desired.

The complete system is illustrated in Fig.1.

4 Refactoring Revisited

4.1 „Wiener Moderne“ to „Exile“

The development of content management and publishing system for this type of open distance learning project turned out to be a phase of multiple refactoring steps. There are many reasons for this process. First of all: the technologies used from the beginning (XML, XSL, template engines) developed very fast, requiring adaptations and re-builds. But more important: the literature projects are among the first projects of this type as well as the cooperation. This required some time to find out the best solutions for the aimed goals and find the right interaction mechanism for the users and for the german literature and language scientists. An easy user interface to enter the data was necessary, the design had to be flexible, and different output formats were required, beeing open for future developments (as described above).

Not only did the technology change, also the size of the second project increased compared to the first project. For example: the second project had the double amount of lectures compared with the first one, additional interactive elements, ...

Just to give a brief quantitative overview about the size of the second project. It contains about 600 articles, 40 lectures, more than 3000 keywords, nearly 5000 multimedia elements (this includes URLs and literature references, but also more than 650 images and about 40 videos and 120 audio files). From this „raw“ data about 40 pdf files and nearly 12.000 html files are generated.

After experiences with the first project, several extensions were desired:

- Adaptions in many parts of the infrastructure to fulfill the needs of the new project.
- New interactive elements, to allow interaction of students with the system.
- A client server architecture for better data management and hence a new user interface to enter the content.
- A new design for the second project.
- Additional Multimedia Elements (video, audio, ...).

4.2 Client / Server Architecture

The first project was based on a desktop database, which had some advantage in terms of rapid prototyping, but big disadvantages in data-safty, security and flexibility. In the new

project a client/server architecture was selected. This allowed the separation of user interface and data, which has many advantages:

- Multi User capability
- Backup of Data is easier and straighforward on the server side.
- Generation of Output is more flexible.
- Update of User Interface is easier, as a clean separation between data and User Interface is done

4.3 Template Engine

When the first project was started, we decided to write an own template mechanism for HTML production, with the requirement, that it is easily possible to use html-stylesheets produced by the designer as template for production of the html output. After finishing the first project several open source projects showed up, that implemented other even more powerful template mechnisms.

The experiences showed, that this approach was very useful for our system, as it allowed easily to cooperate with the designer, who supported the technicians with sample html files, which then could easily be transformed to templates for html production.

On the other hand, our template engine turned out to be not flexible enough for the second project. As beeing "standard" conform was always a design goal, we decided to switch to the Apache Velocity [2] template mechanism.

This library is powerful and flexible. It is not only a "simple" template mechanism that allows to "set variables" in a html template, but even allows to use java objects inside the templates. So the "template itself" can call java classes and e.g. database classes to get data. This allows a manifold use of this system. Lately one of us suggested a system, where the complete html generation process could be template-driven, not application driven. This might be a future extension, or part of a future refactoring step.

4.4 XSL

The development of Extensible Stylesheet Language (XSL) standards [1] and even more important xsl tools between 1999 and 2002 was a significant one. This had consequences considering our new project: Not only all xmp parser libraries had to undergo main updates, we also decided to use XSL formatting objects (FO) language for PDF production. In the first project no useful FO processors were available, so we implemented a multiple step PDF generation using an IBM alphaworks library to generate LaTeX out of XML, then PDF out of LaTeX. Though the results were pretty good, the libraries were flawed and the generation process needed manual work, which was off course not desired.

Currently an open source formatting objects processor is available in the Apache XML project (FOP) [3]. Though FOP is also in an early stage, the functionality is sufficient for our purpose. The consequence is a much better integrated and flexible product as a combination of XSLT and XSL:FO can be used to generate the PDF documents. This is possible automatically without the need to do manual interactions.

4.5 Documentation

Documentation of the new application is done using javadoc for sourcecode documentation and UML (unified modeling language) for a system overview.

4.6 Compatibility between Projects

Refactoring steps often have consequences in terms of compatibility to old data. This is true especially in our case, as we did not only make a refactoring of the initial project applicaiton, but also extended it in many ways. So every new feature had to be reviewed in terms of compatibility with the old data and goals of previous work. This sometimes conflicts with the ideas of the new project manager, and compromises have to be worked out. Currently work is done to update the "old" data and templates for working with the new system.

4.7 Refactoring Steps – in a Nutshell

Especially in research project, when it is not always precisely clear in which direction a system will evolve, it sometimes is a good decision to redo complete systems, or a least big parts of it, also to increase the quality of the design after experiences with previous version. This was what we decided to do after beginning the second project. Nearly every single stone of the first infrastructure was turned and refactored.

The process could be seen as a multi-step approach towards a system useful for many cultural open-distance-learning projects:

- The first project started as a learning phase, not only in technological sense, but also, and maybe most important, in terms of cooperation between different cultures of science.
- In the first & second project there was off course an inception and production phase.
- The start of the second project was used as turning-point. All parts of the system were discussed, following the experiences with the first system, and a complete refactoring was performed as described above. The quality of the new system hence is improved in many ways compared with the first version(s).
- As a (preliminary) last step, additional specific refactoring is desired for a new project allowing to add flexibility in some parts of the publication and fixing several issues in the Client/Server system, which is a new part since the second project.

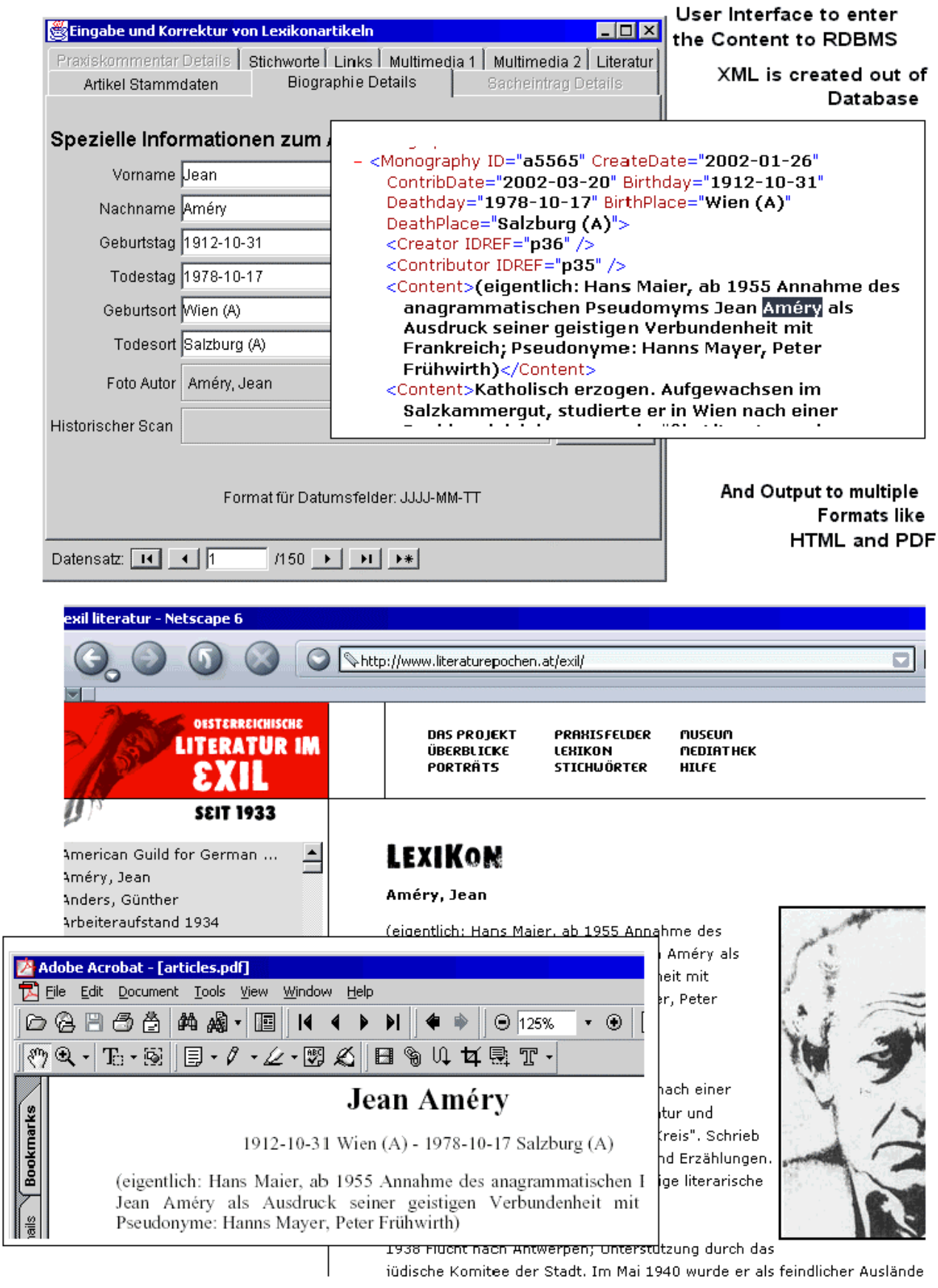

Fig 2. **Illustration of the Publishing Process**

References

1. Adler et al. (2001), XSL (XSLT and XSL:FO W3C Recommendation) http://www.w3.org/TR/xsl/
2. Apache Jakarta Velocity Project: http://jakarta.apache.org/velocity/index.html
3. Apache FOP (Formatting Object Processor) Project: http://xml.apache.org/fop/index.html
4. Bray, Tim and Sperberg-McQueen, C.M. (1996), Initial XML draft, http://www.w3.org/TR/WD-xml-961114.html
5. Connolly, Dan (1997), Extensible Markup Language (XML), http://www.w3.org/XML
   Sun (1999), JavaTM Technology, XML: A Perfect Match, http://java.sun.com/xml/
6. Harold, Elliotte Rusty, XML Bible, IDG Books (1999)
7. Leventhal, Michael and Lewis, David and Fuchs, Matthew, Designing XML Internet Applications, Prentice Hall PTR, Upper Saddle River (1998)
8. Schatten, A. and Tjoa, A Min and Zelewitz, K. and Stockinger, J. (2000), Austrian Literature Moving to Cyberspace - A Framework for Building an Open Distance Learning Website using Platform Independent Standards Like XML, *ED-Media Conference Montreal 2000*