Austrian Literature Moving to Cyberspace – A Framework for Building an Open Distance Learning Website using Platform Independent Standards Like XML

Alexander Schatten, Klaus Zelewitz, A Min Tjoa, Johann Stockinger

Abstract

Experiencing a steadily growing interest in Austrian Literature, we planned to initiate an Open Distance Learning (ODL) course. For that reason, it was desired to build a framework that allows to record the data required for publication in a highly generic form, which is difficult in the case of lecture data and multimedia information. Nevertheless the aim was to separate data from presentation, for not loosing information and gaining flexibility for future use of the data in arbitrary other (scientific) projects.

Moreover as the internet and multimedia technology is evolving with an enormous speed, our approach allows easy adoption of data to future internet or multimedia technology. Data is entered into a database using a GUI for easy access. In the next steps, the database data is transformed into XML. Using XML and DOM/SAX interfaces a flexible generation of different presentation formats like HTML and PDF is automatically possible, as well as the reuse in other context. In the future probably direct use of XML (new browser generation) could be possible using XSL or DSSSL.

A further condition was to "hide" the "difficult" technical background from the german language and literature scientist, to allow easy entering and manipulation of data and multimedia elements.

1. The Background

Close contacts between the department of german language and literature of the Paris Lodron University and an increasing number of German Departments in various CEE-states have been established within the past 25 years. Thus we have experienced a steadily growing interest for the topic „Literatur in der Wiener Moderne", focussing a most interesting, shifting cultural scenery between about 1870 and 1910, comprising both the desires of incoming students and the demands for outgoing Salzburg professors. At the same time it became obvious that it would not be possible to cover those demands by means of physical mobility, not only due to limited finances.

Therefore it seemed more or less evident to produce an online learning package. Our application passed a national funding contest successfully in late 1998. It was the Austrian Ministry of Science and Transport helping to establish co-operation between the Department of german language and literature of Salzburg University and the Department of Software Engineering, Vienna University of Technology, and in early 1999 those two institutions have started their interdisciplinary work on that specific learning package at very good terms.

With various German departments in Austria's CEE-neighbouring states (i.e. in this context: so far partners in Albania, Bulgaria, Croatia, Czech, Hungary, Lithuania, Poland, Romania, Slovenia, Slovakia, Ukraine) we have found a strong interest both in co-authoring and participating in the use of the forthcoming package.

2. Basic Ideas

A set of basic ideas were taken into consideration when starting the co-operation between a team of german language and literature scientists at University Salzburg and a team of technicians in Vienna.

- Entering and collecting information not building layouts first of all.
- Reuse of data for future projects should be easy, so the data produced by the team in-Salzburg should be stored in a highly generic form with a minimum loss of information.
- These “features” were especially important for the colleagues from ministry of science, as future extensions in content and further development should be as easy and consistent as possible.
- Platform independent tools are preferred for maximum flexibility (Java, XML, DOM, JDBC)
- Metadata initiatives like Dublin-Core and RDF should be implemented for easy interoperability with other information systems. Moreover the system should fit in future indices built by next generation search robots (“intelligent software agents”) or bibliographic systems.

Besides this set of conditions and a lot of good ideas some problems existed or appeared while co-operation:

- The team of german language and literature scientists is very interested in web technologies, but nevertheless some original ideas needed to be transformed or modified by technological reasons, like database principles, or because of knowledge that leads to more flexibility in future use, but little more difficulty in entering information at the moment (e.g. XML)
- The two teams were located in two cities (Salzburg and Vienna) so a good concept for information exchange and documentation was needed. Additionally also the colleagues from the ministry of science and from other european countries were interested in information exchange.
- Hence not only results and discussions had to overcome a long distance also technical information and instruction into new software tools or database interfaces.
- Moreover it was an interesting experience to work in an interdisciplinary way, as both teams had sometimes a different point of view. A good communication structure was obligatory to find solutions valuable for both sides.

3. Generic vs. “Layout” Concept

3.1. Conventional Concept

In the Windows- and Office- centric world it is a usual approach to sit down and start writing letters, articles, books and websites. In many cases this “easy” approach is suitable, but sometimes not:

In our project we tried to avoid this method by the following reasons:

- Loss of information.
- Low flexibility: reuse of data is not possible without a huge (manual) effort.
- Layout (design) and content gets mixed up: Keeping a consistent layout in the whole website is a serious issue.
- Extension of content or changes in the layout becomes nearly impossible after completion.

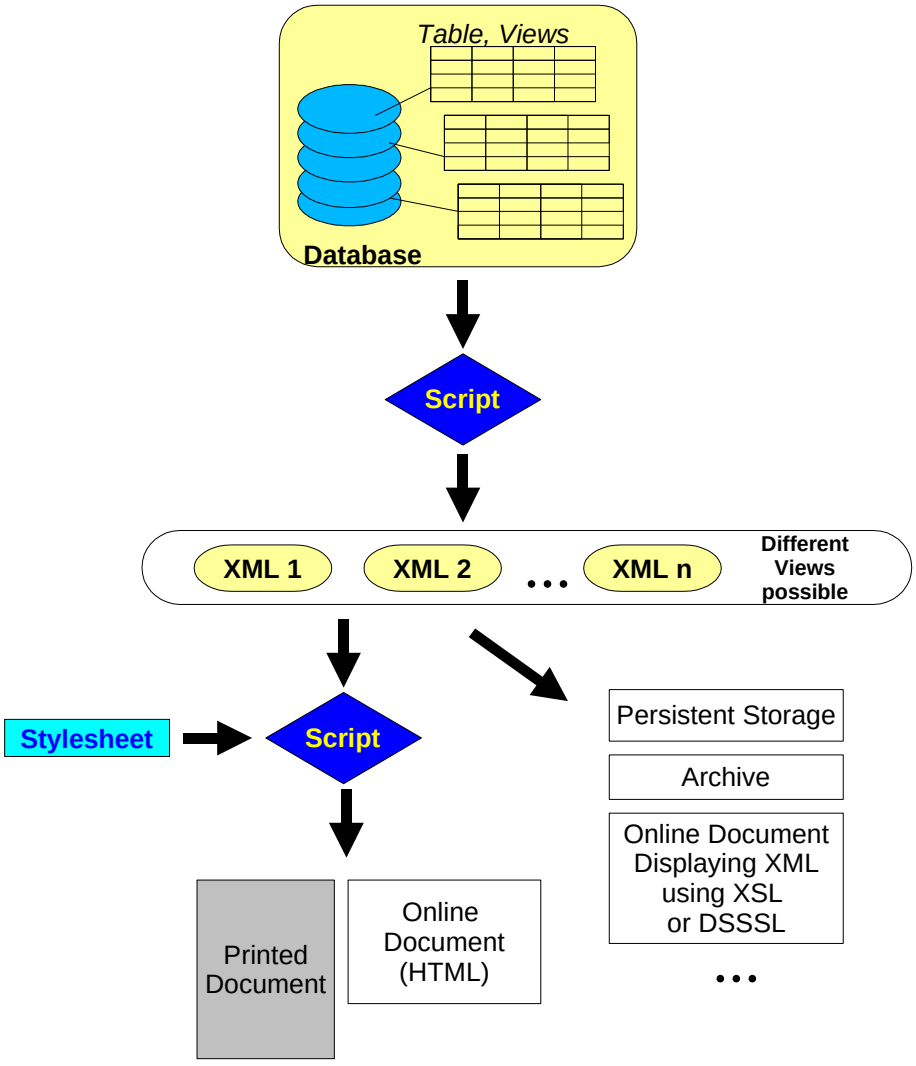

Fig. 1. Generic Concept: From Database to Publishing

3.2. Generic Data Concept

Information is structured and entered into a relational database management system (RDBMS); We try to collect and structure not only on the surface “hard” structured data, but even “soft” data like lecture texts.

Following this approach we avoid the mentioned problems and it is again easier to add meta-information (metadata) like: keywords, subjects, abstracts, links, ... After having stored the data in the database system further processing of the information is conceivable in a flexible way (Fig. 1 illustrates this concept).

By modelling the information using entity-relationship diagrams, redundant information is avoided. This is not only a “nice side effect” for the technicians, but means also more comfort for the team in Salzburg, as smarter entering of data is practicable. This is especially true for the “multimedia components” (like images, music, URLs, ...) as each entity has to be entered precisely one time, regardless how often it will be used.

Moreover particularly these multimedia entities often need maintenance like modifying changed URLs or image filenames and the like. As the access to this data is easily possible through the visual database interface and these entities are stored in a non-redundant way, the change of a URL used many times in the website is one simple change of the database.

The next step is the generation of XML files, which contain the information for publication and the meta-information. The XML data is then used to generate the files for publication (HTML and PDF). The metadata stored in the XML file can be transformed into the desired metadata-format like RDF.

Again we find an advantage of the structured approach: high-quality indices can be built as the information needed is already part of the data. There is no need to use “low quality” full-text retrieval systems.

Furthermore XML is a very flexible format: not only HTML and PDF files can be produced, also data-exchange or direct use of XML using XSL or DSSSL stylesheets is a possibility for the future (the next browser generation should be capable displaying XML directly; Internet Explorer does it already in an experimental way).

4. Data Structures

4.1. Introduction

Besides some “free” elements the project could be divided into a set of main components:

- Biographical Articles
- The “main” information for distant learning: Lectures
- Multimedia Elements
- Original Literature (Citations, Full Text)
- Index

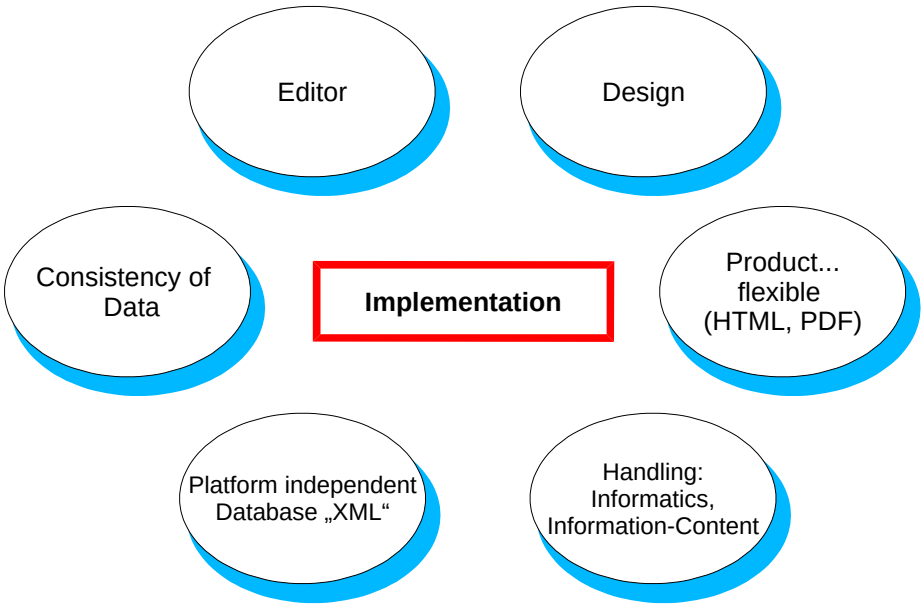

Fig. 2. Different Aspects of Implementation of Lecture Data Structure; Some aspects can be seen antagonistic.

As mentioned above a high generic representation of the information was desired. The database model for the biographical articles and multimedia elements were rather “easy”, as this kind of information fits well into database structure.

Much more difficult was the finding of a suitable structure for the lecture texts.

4.2. Multimedia Entities

The organisation of multimedia-elements is separated into common data used for all multimedia entities like “Title”, “Author”, “Date”, and so on, which is stored in one central table, and the media-specific data (e.g. “URL”, Filenames of Images, ...).

All stored multimedia elements can be used for lexicographic articles as well as for lecture elements, hence have to be entered only once, independent of the number of uses.

4.3. Biographical Articles

The development of the data-structure for biographical articles was relatively simple, as this kind of data is by nature highly structured. At this place the work concentrates to build a suitable data-model and implement this model in the database.

4.4. Lecture Data

Lecture data showed to be rather difficult to structure as it contains textual elements (headers, “normal” paragraphs, citations, tables, ...), multimedia elements and references. Moreover, contrary to the biographical data, the order of the lecture elements have to be stored as it is not evident like in a lexicographic structure (see also Fig. 2).

So several aspects needed to be considered:

- The data should be stored in a “neutral” platform independent way, preferable in XML.

- Consistency of elements and links must be guaranteed.
- Design should be separated from information.
- Entering and editing of data must be as easy as possible as non-technicians should be able to manipulate the data.
- The handling of the data should be as easy and flexible as possible.

Some of these aspects are antagonistic, so the building of the data-model and interface was accompanied by intensive discussions to find a valuable compromise.

We decided to collect the lecture data on the basis of the above mentioned elements. Hence each header, text or multimedia element is entered separately.

One idea was to enter the lecture data directly in the XML format. Certain arguments strongly opposed it:

- XML is yet in a kind of "experimental" stadium, especially as to some important elements like linking. As a consequence there are hardly usable "desktop" tools like XML Editors available.
- References to (multimedia) elements already stored in the database are difficult, or consistency is not guaranteed when entered manually. If stored directly in the database the consistency is guaranteed by referential integrity provided by the database directly.
- The editing of the data should (must) be user-friendly. This is not guaranteed at the current state of XML editors.
- Moreover two different tools would be required for entering data: The database (front-end) and an XML editor. Our solution needs only a uniform database user interface.
- Upgrading this solution to a client/server system for multi-user purpose is difficult. In the case of a "pure" database solution this step is a natural one.

As a consequence of these considerations we decided to store also the lecture data directly in the database, though the building of the database structure and a user-friendly GUI was a challenge.

4.5. "Free Data"

Off course it was not possible to organise all data in a highly structured relational database model. Some elements had to be created "piece for piece".

Nevertheless one essential constraint is, that linking between the automatically generated pages and the "free" written pages is possible without problems and is consistent.

5. Technical Details:

5.1. Database

As database system we decided to use Microsoft Access 97. At the moment there is no need for a multi-user solution also the data is stored in the Access database too.

Nevertheless we tested (at the developer side) a client/server solution, using Access only as Database front-end storing the data in a dedicated RDBMS server. The data access is performed using ODBC in this case.

The database contains currently about 25 tables to store all mentioned data (articles, lectures, multimedia elements, metadata). More technical details can be seen on our website.

5.2. XML

Extended Markup Language XML is a recent W3C recommendation (Bray 1996, Connolly 1996, 1997). XML is like HTML a SGML application. It showed that HTML has a lot of restrictions while SGML is very powerful but rather complex and was initially not designed for online use.

So essentially XML is much more flexible than HTML but the complexity of SGML was avoided. XML allows to create your "own" markup language, which offers the possibility to define exactly the tags necessary to store the desired information. Syntactically XML is similar to HTML but may not used that "sloppy". (E.g. no starting tags without end-tags).

We registered the following advantages of XML for our project:

- XML is a text-format, hence platform and system independent: This is a relevant as *persistent storage* of the information is desired, independent of special database software or other technological developments/changes
- XML offers the possibility to store the project data without loss of information.
- XML is (contrary to HTML) *easy to parse*. Consequently lots of free parsers are yet available, many written in Java.
- It is a human-read- and understandable data format. The consequence is, that "everybody" has access to the complete project data without the burden of reading and interpreting binary data formats. This is especially important to store valuable data for future use.
- The RDBMS system is a good tool to store and handle the project data, but data representation in the RDBMS system is not very good suited for publication purpose. The XML files are designed having the destination publication in mind.

5.3. Publication of the Information

The smartest way of publication would have been using XML directly! This is possible in theory as part of the XML activity are definitions of style-languages. Especially two are in discussion: DSSSL (coming from SGML) and XSL. So the idea is keeping the data without loss of information in XML and publishing this data "through" a style "filter" namely DSSSL or XSL. The problem is, that neither Netscape Navigator not Internet Explorer are capable to perform this task at the moment, but next browser generation (e.g. Gecko rendering engine) should.

Furthermore one other essential element of hypertext, namely linking is currently under heavy discussion, and seems not to be ready for production use at the moment.

The consequence was, that we needed to convert the XML data into HTML specially prepared for online purpose (no long text elements per page, clear design for easy online reading ...) and into PDF for off-line use (printing). For future use it is off course desired to direct-publish the XML data using XSL (?) stylesheets.

Nonetheless we separated information from design: A designer generated design-templates for each part of the project and the scripts automatically created HTML pages get-

ting the information out of the XML data and using the design-templates to build the layout.

5.4. Cross-Linking

As mentioned above cross-linking is indeed a problem as there are several blocks of information like: lexicographic articles, lecture notes, “free data”, and so on. Clear nomenclature for file-naming was fundamental. The problem can be separated into three sections:

1. Linking *to* lexicographic articles
2. Linking *to* lecture elements
3. Linking *from* articles or lecture elements to “free data”

The first point: linking to articles is easily solved, as each article is stored in one HTML file, and the filename is constructed using the unique database ID hence is easily predictable.

The second problem: linking to lecture elements is a more difficult one, as usually more than one lecture element is used in one HTML file. Therefore the links were generated or modified automatically when the lecture data is created, using lookup tables storing filename (and anchor) for each (unique) element ID.

The third point again is rather easy: Also “local” links are handled using URL-multimedia resources, which are checked and modified using the database front-end.

5.5. Conversion Tools

The conversion problem can be divided into (at least) two sub-problems:

1. Generation of XML files out of the database and
2. generation of HTML and/or PDF files out of the XML data.

As mentioned above it seems that Java has adopted XML as “standard” data format, hence lots of free XML parsers of high quality are available. As also the database linking using JDBC (Java Database Connectivity) works reliable, we desired using Java as scripting language.

For generation of XML files, first of all a database connection is required: This connection was performed using JDBC (and ODBC under Windows, native JDBC driver under Linux/mySQL). Having access to the database, the XML library *Project X* from Sun Microsystems (Sun, 1999) was used to build the XML tree. The Project X library then generates the XML data.

The next problem is building the HTML-website. This task is also performed using Java and the Project X library. The XML parser reads the XML files and builds a tree-representation of the data. Using this tree easy access to all parts of the document is conceivable.

Finally the HTML files are generated using HTML-stylesheets to having separated the conversion scripts from design/layout.

5.6. Metadata

The basic idea of metadata is the problem, that Information in the World Wide Web is *human* readable and (hopefully) understandable, but not *machine*-understandable. Why is this a problem?

One of the biggest and yet emerging problems in cyberspace is finding information. Even if we know that the information is available, it is often not easy to locate it. One problem is the enormous size of the WWW, but the other problem is, that search engines, library systems and indices do not *understand* what they can find on the web.

Metadata can be a solution to this problem as it offers a description of the available information. Metadata is *information about information*. The fundamental requirement of metadata is, that it is machine *read*- and *understandable*!

One of the first Metadata Initiatives was the PICS: *Platform for Internet Content Selection*, but was first of all considered to rate content (W3C, 1996). The successor of this first attempt were the *Dublin-Core Metadata Initiative* (Dublin Core, 1997) and the Proposal for the *Resource Description Framework* (RDF) (Lassilla, 1997).

Syntactically RDF is XML based, which is a consistent method designed for future use with XML documents. Nevertheless RDF can be applied to HTML documents too.

Especially in a project where lots of structured information (lexical data, lecture text elements) are available, clear usage of metadata is a wise decision as we hope to be fit for future implementations and developments of electronic library systems.

Also content rating comparable to PICS and providing copyright information is possible using RDF, which is though not that relevant for our work.

But the main aspect will be the knowledge exchange using intelligent software agents. The technology of “intelligent” agents is still a research topic, but will emerge soon as a relevant method for information retrieval in my opinion. As far as we can expect now, RDF will provide the “food” for these software tools.

Moreover the creation of metadata is not a real burden, as our information is well structured in a database system already containing the needed metadata (keywords, subjects, abstracts).

6. Co-operation Details

One basic element of the project co-operation was the fact that the teams were located in different cities (Salzburg, Vienna). Moreover the colleagues from ministry of science participated actively and even the colleagues in Vienna worked at three different locations (Ministry, University of Technology, Designer).

The consequence was, that extensive use of all communication channels was necessary. Off course telephone and personal meetings were relevant aspects, but even more important was constant discussions using e-mail and three special internal websites.

The german language and literature team in Salzburg permanently documented their work and ideas at their website in Salzburg, the software team provided a documentation website offering all resources, manuals and technical documentation, last but not least also the designer showed the last concepts and ideas on his internal pages.

Much more than usual in other projects constant and hence permanently updated documentation of the current work was prerequisite for this kind of co-operation.

7. Future Development

We all are positive that already existing well functioning physical academic mobility (SOCRATES, CEEPUS and others) and cross border co-operations will be enlarged and improved by strong virtual supply.

Nevertheless this is one of the first steps in this direction. Further german (Austrian) literature projects are planned. Not only adding new content will be part of the future development, also update in new technological directions are desired. Particularly the next browser generation and their ability to directly render XML will influence the future development from the technological side.

Regardless of the direction the internet is developing, we hope to be on the "safe side", as data is stored and processed using open standards. So at least the core of the project, namely the basic information provided by the german literature scientists should survive the multimedia future.

8. Go To "Literatur in der Wiener Moderne"

Http://... not available for public use at the moment because we are heavily working on this site.. The correct address will be added in final release of this document.

For further interests in this project please send an e-mail to lwm.interesse@sbg.ac.at

9. References

- Bray, Tim and Sperberg-McQueen, C.M. (1996), Initial XML draft, http://www.w3.org/TR/WD-xml-961114.html
- Connolly, Dan (1996), Extensible Markup Language (XMLTM) Activity, http://www.w3.org/XML/Activity.html
- Connolly, Dan (1997), Extensible Markup Language (XMLTM), http://www.w3.org/XML
- Dublin Core (1997), Dublin Core Metadata Initiative http://purl.org/dc
- Lassila (1997), Introduction to RDF Metadata, W3C Note, 1997, http://www.w3.org/TR/NOTE-rdf-simple-intro-971113.html
- Sun (1999), JavaTM Technology, XML: A Perfect Match, http://java.sun.com/xml/
- W3C (1996), PICS Label Distribution Label Syntax and Communication Protocols, Version 1.1, W3C Recommendation, 1996; http://www.w3.org/TR/REC-PICS-labels