

# The First Steps in Designing an SGML-Based Infrastructure for Document Handling

Tone Irene Sandahl\* & Astrid E. Jenssen\*\*

\*Department of Informatics, Box 1080 Blindern, N - 0316 Oslo, Norway

\*\*Center for Information Technology Services (USIT), Box 1059 Blindern, N - 0316

University of Oslo, Oslo, Norway

tone.sandahl@ifi.uio.no

astrid.jenssen@usit.uio.no

## Abstract

*In this paper we present and discuss efforts to design an SGML-based infrastructure in an institutionalized work setting. The initial goal was to improve the "functionality" of documents in order to access, update, search, use and reuse, retrieve, present, exchange and distribute them independently of time and place and without loss of information. From a technical point of view, standardization is a prerequisite for success. In the pilot project presented in this paper, Standard Generalized Markup Language (SGML, ISO 8879) is applied. However, the study has shown that writers experience standardization as a restricting factor in their work, and local flexibility is essential. In relation to SGML-based infrastructures, this has consequences for DTD design and for the selection of authoring tools. The study has also indicated that documents are artifacts that are integrated into practice, and this has to be taken into account in design of document technology.*

Key words: *SGML-based infrastructure, DTD-design, documents in work*

## 1. Introduction

Since early on in the computer age, there has been a need to classify, compute, combine and recombine, count, sort, and manipulate information. From the late 60s, this has been done by cutting the information into little pieces (data) and putting it into databases. Databases are fine for discrete, predictable pieces of information, but they do not work very well for information like stanzas, scientific descriptions, or maintenance procedures (Alschuler 1995). A database system is basically a computerized record-keeping system (Date 86). As stated by Reinhard (1994), at least 80% of electronic information in organizations is in

the form of documents, as opposed to database records. Traditionally, documents have been static, represented as files on disks. Until PCs were networked, these files usually belonged to only one user and passed from one person to another in printed form. It is a challenge to make these documents, or files, more dynamic, in order to be able to access, search, use and reuse, retrieve, present, exchange and distribute them without loss of information (Reinhard 1994).

This paper presents and discusses the efforts involved in the first steps of designing an SGML-based infrastructure, and considers the complexity of this. The concept of an infrastructure denotes all the documents and practice required to support people adequately in carrying out their work (Jewett & Kling 1991, Star & Ruhleder 1994). The concept of an SGML-based infrastructure indicates that the technical solutions are based on SGML. From a technical perspective, application of SGML is an appropriate approach for three main reasons: i) Standardization is necessary for infrastructures to exist; it is the technical backbone (Hanseth *et al.* 1996). ii) SGML makes it possible to use structured concepts in text in general, which lessens the difference between documents and databases, as well as improving search and retrieval in texts. iii) SGML is independent of software, systems and presentations, and it supports "open systems".

At the end of 1992, a project was initiated at the University Center for Information Technology Services (USIT) to determine what type of electronic infrastructure could deal successfully with electronic documents and other forms of information at the university. The infra-

structure had to address the whole life cycle of a document, i.e., production, updating, filing, administration, distribution, presentation and reuse. The use of SGML in the project emerged as a possible key tool for describing the documents and their content.

In 1993, to gain experience in the practical use of SGML for some parts of the information produced at the university, USIT established a pilot project that involved developing an open and flexible solution for the production, exchange and distribution of the university's course catalog. The pilot project was initiated to develop a technical infrastructure and administrative routines for dealing with the catalog, which contains dynamic information and "new functionality" for the students and the staff. A detailed presentation of the development process itself can be seen in Jenssen & Sandahl (1996).

The paper is structured as follows: In Section 2, a short introduction to SGML is presented, followed by the research approach in Section 3. Section 4 presents the practice of catalog work, while Section 5 briefly presents the design and implementation. The evaluation of the pilot project is in Section 6, the discussion in Section 7, and finally the conclusion in Section 8.

## **2. Standard Generalized Markup Language (SGML)**

There are at least two different ways to achieve information interchangeability between systems: standardization on applications, so that the applications can work on each other's information, or standardization of the information itself,

so that it can be processed by any application. SGML supports the latter solution.

SGML is designed to enable text interchange, and it is intended for use in the publishing field (Smith 1992). Since then, it has been increasingly adopted as the international standard for data and document interchange in open system environments, including the automotive, defense, commercial aerospace, pharmaceutical, electronics, and telecommunications industries (Van Herwijnen 1993).

A basic design goal of SGML is to ensure that information encoded according to its provisions should be portable from one hardware and software environment to another without loss of information (Goldfarb 1992). The basic idea is very simple: the text is described in terms of its structural components (descriptive markup) rather than its presentation (procedural markup) in a single medium.

A descriptive markup system uses markup codes, which provide names to categorize parts of a document. Markup codes such as <course> identify a portion of a document and assert of it that “the following item is a course”. All the text is coded as plain text. SGML thus enables the interchange of text across platforms, because there is no need for “translation” to meet hardware requirements. The same document can readily be processed by different types of software, each of which can apply different processing instructions to those parts of it which are considered relevant. In addition, different sorts of processing instructions can be associated with the same parts of the file. Since only the structure and/or content of a document are marked, a given viewer of that docu-

ment can decide what the “look” and “use” will be. The markup of the document never changes—only the way it is interpreted.

SGML supports the notion of a document type, and hence a “Document Type Definition” (DTD). An SGML document always has an associated DTD that specifies the rules of the model of the document; for example a DTD for a course catalog might specify that the document type (catalog) must have information about one or more courses. Furthermore, each course must have a code, a title, an optional description (descrip), followed by zero or more combinations of day, time and place. Information about lecturer(s) is required, and so forth. The type of a document is formally defined by its parts (course, day, time, place, ...) and their structure in the DTD. Figure 1 shows an example of the syntax of part of a DTD.

---

**FIGURE 1. Part of a Catalog DTD**

```
<!ELEMENT catalog - -  
  (course+)>  
<!ELEMENT course - - (code,  
  title, descrip?, (day,  
  time, place)*, lecturer+)>  
<!ELEMENT teacher - - (fname,  
  sname, email?, phone*,  
  fax?)>
```

The users of the standard design the DTD; hence, the DTD is not predefined in the standard itself.

---

### 3. Research Approach

The research approach was pragmatic, as described below.

First, we decided to use the university study catalog as a pilot document. The goal of the pilot project was stated before the project start, namely to make the catalog electronic and structured in order to improve its production, exchange, distribution and application. SGML is intended to support this.

We entered the problem environment, analyzed the work situation, and identified roles related to the production, exchange, distribution, and application of the catalog. We took part in the design process and evaluated it continuously. After finishing the pilot project, we reflected on and recorded what we had experienced during the process, which is documented in this paper.

### *3.1. The Catalog as a Pilot Project*

The catalog was chosen as a pilot project for several reasons. The university catalog is well known to the students and staff at the university, and many people use it when carrying out their jobs. The production process of the catalog is attractive in relation to other crucial documents; the information is supplied from different units (faculties, departments and central administration) at the university and a number of writers are responsible for updating different parts of the catalog.

The electronic version of the catalog facilitates advanced search, reuse of information and presentation on different media (printed, electronically on screen), as well as possibilities for links to other relevant information such as university/faculty regulations, syllabi, curricula, and so on. From a more technical perspective, the information was seen as structured, so it was well suited for building SGML applications.

The main goals of the pilot project were to produce a better (and in the long run, cheaper) catalog, to make it easier to update and maintain, and to gain practical experience introducing SGML at the university. To produce a better catalog involved developing a more readable catalog, making it electronically accessible and adding some new functionality.

### *3.2. The Idea of a Structured Catalog*

As we saw it, a critical aim was to get a structured and well-defined catalog in order to develop new services based on the information in the catalog. Later, we wanted to develop scripts that could manipulate the catalog and its information elements depending on existing situations and products. Services like room allocation, in which a writer allocating a room could get suggestions for a place and time based on existing information in the catalog, as well as advanced search in the catalog information and customize publishing, were meant to be implemented.

A basic principle was that the writers should have to enter information only once. As a result, updating and maintenance of information across presentations and products should be easier, and redundancy could be avoided. If the information was presented in a different setting, the computer would do this automatically. The writers would still have to collect information from lecturers and others, and enter it in the catalog.

### *3.3. The People Involved*

The pilot project was organized as a project group responsible to a steering committee. Both groups consisted mainly of staff from USIT. The project group consisted of 3–5 IT people with 1–2

working full-time, the others working part-time on the pilot. The group was responsible for the system development process. The central administration was responsible for editing the catalog and some of the writing. Each semester, there were about 40 writers from the different university faculties and departments, and several of them worked with the catalog for more than one semester. The writers were located in all the administrations and secretariats at the faculties, the central administration and the study department. In the faculties and the study department, the secretaries or administrative consultants are the writers, where the advisers do the writing at the department level. However, the production of the catalog is only a small part of their daily work.

#### *3.4. Evaluation Methods*

During the project we interviewed 22 different people involved in the project, representing writers and management people at USIT and the central administration unit. Some were interviewed several times. We analyzed 393 email messages sent to a distribution list. We reviewed a number of questions and problems from telephone calls and direct mail. We analyzed minutes of 13 meetings with the writers and more than 20 internal meetings dealing with more technical problems. In addition, we analyzed 3 project reports.

### **4. The Practice of Catalog Work**

The work with, and the use of, the catalog were manifested in the ongoing work procedures. The writers' work, the coordination, and the merging of the catalog

had been done in the same way for many years. In this section we describe how the work was done before the pilot project.

The catalog is separated into sections for each faculty, describing courses offered by that faculty, and different sections for other kinds of courses such as distance education, information about student services, and collaborating institutions. The writers at the central administration maintain information about all sections except the information from the faculties. They have to coordinate with other units at the university to collect information to be presented in the catalog. The writers at the faculty level maintain information common to a faculty, and the writers in the subordinate departments maintain information mainly about courses offered by the department. In addition to writing the text for the catalog, the writers at the faculty level are responsible for making sure that all the text from each underlying department "fits together", e.g. that registration deadlines for classes are correct in relation to the dates set by the faculty, and that there is no overbooking of lecture rooms.

However, all the departments and units are responsible for their parts of the catalog, for collecting information and for distributing updated information to the students and staff. For instance, the writers at department level are in contact with different lecturers to collect information about which courses are to be offered each semester. They have to coordinate the allocation of lecture rooms, taking into account the lecturers' preferences for the day, time and place and, in cooperation with the writers at faculty level, to avoid overbooking. They do the updates and distribute the results to the different lecturers for proofreading.

**TABLE 1.** The work done at the different stages of the system development process

<i>Work period</i>	<i>Work done</i>	<i>By whom</i>
Catalog 1 93: March-June	Document analysis	USIT/writers
	Developing the first version of the DTD	USIT/writers
	Manually encoding the information for the autumn catalog	USIT
	Developing the printed version	USIT
	Developing scripts for conversion to HTML	USIT
	Setting up the catalogs, files and access in the Unix file system	USIT
	Organizing the work flow	USIT
	User training	USIT
	User support	USIT
	Evaluation	USIT/writers
Catalog 2 93: July-December	Developing the second version of the DTD	USIT/writers
	Merging already encoded information into the new version of the DTD	USIT
	Encoding the information using templates in word processors	Writers
	Conversion of new information to SGML	USIT
	Improving the printed version	USIT
	Printout possibilities for the writers through Unix	USIT
	Organizing the Unix file system	USIT
	Organizing the work flow	USIT/writers
	User training	USIT
	User support	USIT
Evaluation	USIT/writers	
Catalog 3 94: January-June	Improving the second version of the DTD	USIT/writers
	Improving the electronic version	USIT
	Using the SGML editor for updating the information	Writers
	Developing schemes to be used with the SGML editor	USIT
	Improving the conversion to HTML	USIT
	Organizing the Unix file system	USIT
	Organizing the work flow	USIT/writers
	User training	USIT
	User support	USIT
	Evaluation	USIT/writers
Catalog 4 94: July-December	Further improvements of the second version of the DTD	USIT/writers
	Using SGML editor for updating the information	USIT
	Improving the style sheet used by the SGML editor	Writers
	Organizing the Unix file system	USIT
	Organizing the work flow	USIT
	User training	USIT/writers
User support	USIT	
Evaluation (interviewing)	USIT	
95: January->	The system in ordinary use	USIT/writers
	Evaluation	USIT/Dep. of Infor- matics/ writers

Then they update the information again as often as necessary. If students need to be informed about important changes after the catalog is sent to the printing office, e.g. changes in class time scheduling, the writers make an A4 document that they display on boards outside their office, students' lecture rooms, and workplaces.

Different departments and faculties presented the same information in the catalog (e.g. information about courses, time and the places for classes) somewhat differently. For instance, the mathematics presented course information in a tabular form, while the law faculty presented the same information as free text.

The writers used word processors such as MS Word or WordPerfect. They were told which font and font size to use in headings, paragraphs and so on, but no style sheets were available. They saved the documents on their own computers, and paper copies were exchanged or distributed during the production process by mail. Much of the updating work was done on paper versions.

When the departments and faculties were finished with their work, they all sent the files or documents on diskettes to the central administration unit. At the central unit, the final proofreading was done, and references and indexes were created manually. The administration unit sent the catalog on diskettes to the print office. Even before the catalog came back from the printing office, some of the information was outdated due to last-minute changes in course schedules, etc. Hence, there was a need to update information until just before the publication date, and a continuous need for updating of further changes. The writers or the central administration contacted the

printing office directly to have them incorporate the changes in the final printed version.

We observed that some information elements in the catalog were also part of other important handbooks, brochures and catalogs at the university. There were three or four original versions of the information elements, and it was hard to avoid inconsistency. The catalog was published only on paper. It was about 450 pages long, and 50,000 copies were printed twice a year. These were available to students and administrative staff at the university as well as other educational institutions. It was fairly expensive. Electronic services and services such as customizing publishing and publishing on demand could save both paper and money.

## 5. The Design and Implementation

In the pilot project period, the following were emphasized in design: the DTD design, the editing environment, the printing environment, WWW presentations, an environment that supports the interdependence in work to some extent, and finally training programs and support. A summary of the process is presented in the table below. However, a more detailed presentation of the implementation can be seen in Jenssen & Sandahl (1996).

### 5.1. DTD Design

The first version of the DTD was developed through document analysis led by the system developers involving the writers, managers and people from the central administrative unit. The intention was to make the catalogs' structure rich enough to allow retrieval of information

directly from databases, links to other information and functions for presenting different views of some parts of the information in the catalog. As a result of the way that the conversion routines were programmed for the printouts, some elements were also included in the DTD to ensure that the layout in the paper version was correct and attractive. For example, for information about courses, there were different elements to be used depending on the day information, such as `<day>` for one day, `<dayint>` for day interval and `<dayoppr>` for specifying a selection of days: `<day>Monday</day>`, `<dayint>Monday-Thursday</dayint>` and `<dayoppr>Monday, Tuesday, Friday</dayoppr>`. Later, the DTD and the conversion routines for making printouts were changed to use only one element for information about days, and still ensure an appropriate layout.

The focus on technical solutions and products such as printed and electronic versions led to a DTD that contained a large number of elements for the users to deal with in the writing process. A DTD that seems appropriate for the technological solution may not be appropriate for the writers. On the basis of practical experience, the DTD was reorganized and improved during the project. It became less detailed for the whole catalog, removing some elements and generalizing others, but was still rich and strict.

### 5.2. *Editing*

A goal specified for the project was to have the writers produce SGML documents according to the relevant DTD. Because of the importance of correct input, using a native SGML editor was seen as the appropriate solution for doing

this. An SGML editor is context-sensitive. It knows the predefined structures defined in the DTD. It may incorporate a validating parser that makes it possible to avoid markup errors and guarantee that the document is structurally correct. The editor Author/Editor was applied, one of the reasons being that it was one of the few tools available for the PC, Macintosh and Unix platforms at that time. The editor is a tool for focusing on the document's content and structure. It has some functionality for adding different layouts according to the structure, but this is not adequate to fulfill WYSIWYG layout requirements according to the printed version of the catalog.

Use of the editor was integrated stepwise. For the first catalog, the information from the writers was manually encoded using SGML editors and other tools by a group of people at USIT. The second catalog was produced by the writers using word processors and style information, and then further structured and converted to SGML by USIT. For the third and subsequent catalogs, all the writers have used the SGML editor.

### 5.3. *Printouts*

The university had previously developed a print spooling system (PRISS), making it possible to print any file from any computer (Macintosh, PC, Unix Workstation) to any printer on the network. PRISS was applied to get printouts of the catalog on the writer's (or other) local printer with the same layout as the final catalog. The layout was an improvement on earlier versions created before the pilot project; for example, it was consistent throughout the catalog, it included a table of contents for each faculty, and in-

troduced symbols in the margins to highlight important information.

TeX/LaTeX (Knuth 1984, Lamport 1986) was used as the tool for generating postscript files in order to typeset the catalog on paper with an appropriate layout. Scripts were made to convert the SGML DTD to TeX/LaTeX.

The writers order printouts by using WWW interfaces prepared for them, making it possible to print different parts of the catalog to their local printer.

#### 5.4. *World Wide Web Presentations*

To make the catalog available through the WWW, scripts for conversion from the SGML DTD to HTML were developed. The requirements considered for the implementation included presenting all the information in the printed catalog, making it possible to do dynamic updates, and making the result available for users of different WWW clients, e.g. for blind and visually impaired people.

#### 5.5. *Interdependence in Work*

The coordination between the writers at the departmental level and the faculty became more "electronic". All writers have read and write access from their Macintosh or PC to their own catalog file(s), and read access to the other files. Server/Client technology is applied. From their desktop computers, they establish a connection to their catalog on the common Unix server, which is used to manage the different files and the access to them.

The writers leave their files on the servers. When the deadline expires, scripts merge the files to create one common catalog. This is sent to each writer's printer to produce a printout for final review and approval before an electronic

version is created and the catalog is sent to the printing office.

#### 5.6. *Training and Support*

Training and support was emphasized. Writers needed to learn about the structure of the DTD, and the Author/Editor. Twice a year the writers were invited to a two-day course covering both the structure and the editor. An email list and an "SGML phone number" were established for ongoing questions and comments from all the involved participants. The developers also visited the writers at their offices when needed.

## 6. **The Evaluation of the Pilot Project**

This section presents our empirical data from the evaluation of the pilot project. As shown in Table 1, the evaluation was a continuous process during the pilot project. The empirical data from interviews, emails, reports, telephone calls, direct mails and meetings form the basis for this section. The categories presented are based on the main issues raised by writers during the evaluation. The quotations below are from the interviews.

#### 6.1. *The Writing of the Catalog Production*

During the pilot project, the number of writers grew and their work with the catalog production changed. The catalog was still a product, but use of the new technology changed the process. Before the pilot, several of the writers were primarily concerned with collecting information, and the typing was done at the faculty level. This changed to include the responsibility for direct data entry into the system as well.

During the pilot project, the process of producing the catalog was more time-consuming for the writers than before the introduction of the new infrastructure. Some of them had a lighter workload, and ended up as “experts” on the catalog. Others had a longer working day because of the additional demands. Still others received new work tasks, concerned with production of different kinds of information for distribution.

I do another job now. I have been on courses, and spent a lot of time to become qualified to do my job. In fact, I should get paid more now (laughs).

The use of SGML requires discipline in the way text is written. Structuring the information according to a given definition of the document type creates constraints on dealing with it. Usually, people can present their information in their own way by using the tools they prefer. With the SGML, this freedom is restricted.

It is problematic with SGML, because you have to be so damned correct, otherwise you get problems with your printouts. A few “typos”, and then chaos. This is no problem in other word processors that I know. OK, you see the misprint on the paper, but you can read it, and use it!

Some writers pointed out that the freedom to use a well-known word processor, and to present the information in your own way, was gone. One said that he had the feeling of going back 10 years in time, dealing with text markup in editors like RUNOFF. The SGML editor used does not have the same functionality as word processors such as MS Word and WordPerfect. An SGML editor is an assistant for the writer doing markup according to the predefined document defini-

tion. It may incorporate a validating parser that makes it possible to avoid markup errors and guarantee that the document is structurally correct. Some writers stated that when working with the SGML editor they had to concentrate more on the technology and the structure than on the text itself. For most of the text, however, they needed only to fill in information in the right places.

When using Word you almost forget that you are using a computer; it is just there—a tool, which is incorporated in my work. When using the SGML editor I have to think about how to use it – how to include which element, and so on. But I believe I will get used to it (laughs).

The writers were confused by the difference between the logical structure (represented by markup in the text) and the physical structure (or layout, presenting the catalog on paper). We received many questions related to the use of the logical structure. At the beginning, almost every writer related the logical markup directly to the printed catalog. Knowing how a specific markup in a context would look on paper, they used this markup for layout rather than for its logical meaning. For example, they wanted to use the element <emphasize> to mark up a title instead of using the element <title> in the appropriate context for this purpose. Problems related to the differences between logical and physical structure were felt strongly in the beginning of the project. The email concerned with these questions diminished during the pilot project.

## 6.2. *The Need for Local Flexibility*

There was a great deal of disagreement among the writers on how to structure the catalog, and what information ele-

ments should be required. The first design of the DTD was meant to be based on requests and proposals by the writers, managers and designers. However, the writers themselves did not have the same requirements for the DTD. On the contrary, they had conflicting requests and the proposals differed widely among departments, and between departments and faculties and the central administrative unit.

No, I do not agree with anyone! In a way I am happy to have this opportunity to change the catalog. It should have been done years ago. And I see the potential of getting the catalog into SGML; you know ... save money on printing, and you know ... Web and all this stuff. However, we will never totally agree on a common structure for the catalog. The faculties and the departments are too different for that. I can spend hours telling you the structure and content I want, but it will, for sure, be in conflict with what she (a writer from another department) wants.

How can anybody expect us to want to structure our information as a table? We are not the Department of Mathematics. We like to write and read prose. (Laughs). No, I do not have any rational reasons for that. (Laughs).

We experienced unwillingness among some of the writers to change their way of structuring and presenting the information. The writers had strong opinions on a detailed level about their own information. For instance, some wanted to have the general information about a course presented before other information such as day, time, place and lecturer. Others felt that it was important to have the general information at the end of the information about the course.

### 6.3. *Some organizational aspects*

There was a heavy workload for all the people involved in the process of introducing SGML at the university. It was time-consuming to develop both technical solutions and administrative routines. The project report documented over 1200 hours overtime for the technical staff related only to the first edition. It decreased to less than 700 hours for the second edition, and decreased further with later editions.

As mentioned previously, the role of several of the writers extended from pure information-gathering to include updating the system and proofreading as well. In addition, the administration of much of the catalog production process was shifted from the central administration unit to USIT. The technical solution resulted in new deadlines for updates, approval of the catalog and delivery to the printing office. This led to a shift where USIT set the agenda for the writers by dealing with deadlines and organizing courses. USIT assumed responsibility for the graphic representation of the layout for both the printed and the electronic version of the catalog. The editorship was and still is the responsibility of the central administration unit.

### 6.4. *Training and Support*

All of the writers had to participate in training programs, and they all needed time to understand the underlying structure (DTD), to learn the SGML editor, and to get an understanding of how the integration of SGML might influence their work situation. The writers stated that they needed to learn and understand the SGML in order to work with it.

I see the SGML people as a kind of a

doctor for my information. They say that I have to mark it up to gain some new functionality. Of course, I will do that if I know why I have to do it. Comparing it to medicine – I take my medicine if my doctor tells me why I have to do so. I do not take medicine if the doctor cannot give me an appropriate reason. Obvious!

They also wanted to know the benefits of using SGML. They emphasized the need to know the main structure of the DTD, and the where and how of adding new information to the document. Knowing the structure of the DTD requires some understanding about what a logical structure is, and the ability to distinguish between the logical and physical structure. This took time to achieve, but it gradually evolved.

Despite the scheduled training, the writers needed access to some form of help all the time. They needed help to solve technical problems and to figure out what to do with the different parts of the information, how to code and where to put the markup. The interviews, email, and minutes from meetings show that the writers saw the training and support as highly important and necessary, and the many questions from the writers related to the process show that there were obvious reasons for emphasizing support.

#### 6.5. *The Catalog as a Product*

The WWW version of the catalog became very much an electronic presentation of the printed version with the same sequence of the main structure elements, adding some new functionality for searching and navigation through listings of parts of the information. At the early stage of the pilot, very few of the writers were familiar with using the WWW, and they were mainly concerned

about the printed version of the catalog. They primarily used the printed version as a tool in their work with student services.

After the deadline for the printed version, only a few writers took advantage of the possibilities to update the WWW version continuously. Some writers as well as some managers were concerned about which presentation of the catalog should be used as the reference: the paper version or the WWW version. There was no overall agreement from the organization on this subject.

Writers emphasized that the catalog had for years been a kind of contract between the departments and the students. The departments demand that the students read (parts of) the catalog and that they follow the information provided there. On the other hand, the students use the catalog as documentation for what they need to know.

Before, the catalog was a kind of a contract between the students and us, and we wanted it to be like this. How will this be when the catalog changes all the time?

They also mentioned the fact that students could bypass information by not clicking on links to it. Some writers feared that information they saw as important would be less visible on the WWW than in the paper version, and some feared that others would update the information. The writers are very much aware of their role in giving the students the right information. If the catalog does not contain enough information, or the students do not find the information they need, they ask the writers and others in the administrative units. In the printed catalog this information is represented in different “visible” chapters. In WWW

they saw this information as “hidden” behind links.

## 7. Discussion

In this section we point out different aspects that demand serious consideration during the design of SGML-based infrastructures. The discussion is based on empirical data presented in previous sections, and seen from the perspective of system development. In relation to the main goals for the pilot stated in Section 3.1, the discussion focuses on the writing process and the catalog in use, since these are the main issues that the participants emphasized during the evaluation.

### 7.1. Focus on Structure Impacts Flexibility in Writing

There is a conflict between the requirements of a strict DTD and flexibility when authoring. As stated in the sections on empirical work, the writers felt that their freedom to write was restricted because of the editor and the underlying strict DTD. When writers use an SGML editor, they have to be aware of the predefined structure, and they have to deal directly with it when authoring. On the other hand, when using an SGML editor it is fairly easy to avoid ambiguity in markup at input. The editor displays only those elements that are valid in a particular context, which helps the writer to choose the right element. The SGML-encoded documents are ready for further use, handling and management without any conversion or other forms of adaptation. A more flexible DTD may offer the writers a more flexible writing process, but further use of the SGML-encoded document may be restricted, depending

on the degree of functionality, (re)usability and the presentation of (components of) the documents required (Maler & Andaloussi 1995).

Each department presents some general information to the students before the listing of the different courses offered by the department. This might be information about important dates, student services, services for disabled students, and so forth. This should be structured and presented in the same way for all the departments, making it possible to develop services based on the structure. In the DTD, the required structure and content are specified. To fill in the information, the writers must use the correct markup for the different kinds of information and be aware of the predefined structure.

As stated in Section 6.3, the writers have reasons for rejecting the common structure. They could not see the point of making the structure common to all departments and faculties.

There is a tension between the concepts of local flexibility and “global” standardization. The standardization is necessary for communication over networks, and for automation of processes, like merging pieces of information into a catalog and publishing it on paper or the WWW, and advanced search through the catalog information. The tension between standardization and flexibility is also observed by Hanseth *et al.* (1996).

A solution to the conflict could be to offer writers other tools, free them from doing the coding, and instead absorb the cost of conversion into SGML. An obvious approach would be to use a WYSIWYG word processor and appropriate templates (Van Herwijnen 1993). An argument for the use of WYSIWYG word processors and templates from the writ-

ers' point of view is that they can use the word processor they know well, and they deal with layout instead of logical markup. The change in their work situation may be smaller when a WYSIWYG word processor is applied, since the WYSIWYG word processor is already integrated into their work practice. However, we see some drawbacks as well. Studies have shown that users of WYSIWYG word processors do not necessarily apply the templates available, or template styles may be used incorrectly (Sørgaard & Sandahl 1997). Templates that consist of many different styles are difficult to manage. The style list gets long, and which style to choose next is not necessarily obvious. These kinds of problems may force a lack of standardization that *may* introduce other problems - in interactions, merging, and presentation of the information.

As we see it, the choice of an SGML editor or a WYSIWYG word processor should be based on the complexity of the DTD at hand. If the DTD is small with few and understandable elements, a WYSIWYG word processor is preferable. Where the DTD is relatively complex, the effort of learning an SGML editor can be worthwhile, since the WYSIWYG word processor has no mechanisms for managing large sets of styles. In Braa and Sandahl (1998), different approaches to standardization of documents are discussed further.

Another way to solve this problem of restrictions on writing is to make more flexible and/or smaller DTDs that are tailored to suit each of the departments or faculties. To make the DTD more flexible, strong expressions like 'the information about *day* has to come before the *time*, which must be followed by *place*'

can be replaced by expressions like 'the information about *day*, *time* and *place* must be present, but the sequence is optional'.

At the University, none of the writers use all the elements defined in the DTD. All of them use only a subset. For greater clarity, a DTD tailored for each department or faculty can be developed. In this case, all the elements that are not in use are stripped, and not visible to the writers at all.

However, these solutions have to be weighed against technical requirements and the effort of maintenance. DTDs are subject to change, and it is obvious that changing one DTD is simpler than changing 10 or 20. For a programmer, it is easier to make presentations on paper and the Web, or to prepare solutions for advanced search, if the DTD is strict. With a highly flexible DTD the programming is far more complex, because of all the alternatives that have to be taken into account. From a user perspective, the DTD should be as flexible as possible. On the other hand, from a technical perspective, the DTD should be as strict as possible. Where the line is drawn depends on the situation. However, this decision is of vital importance for the success of the SGML system.

## 7.2. *The Catalog in Use*

From a rational point of view the case for an electronic catalog is more or less obvious: it keeps information more accurate, complete and up-to-date, improves portability and makes information less complex and less disorienting (Ventura 1988). However, paper documents carry an aura of authenticity and legality that is difficult to dispel from peoples' minds (Berry and Goulde 1994).

Some writers were unwilling to accept electronic documents as a substitute for “the real thing”, especially at the beginning before they saw them for the first time. There were several reasons for this. People feared that information they considered important would become less visible in WWW than in the paper version. For instance, the information on the first pages in the printed catalog is regarded as important. The main concern of the central administration unit is the information to new students. To reduce the number of inquiries, they want students to read the information before they contact the department. The central administration unit was afraid that the student would click directly into the information about courses, and not see the other information. However, it is a challenge to present information on the WWW in a way that takes this into account.

The catalog is a contract, which is a genre of organizational communication between the students and the staff at the university (Yates & Orlikowski 1992). As pointed out in the empirical data, the departments demand that the students read (parts of) the catalog and follow the information given there. On the other hand, students use the catalog as documentation for what they need to know. The catalog links the staff and the students together; it is necessary for the functioning of the whole university system. The writers experience a contradiction in the meaning between the catalog on paper and the catalog on WWW, because of the change from static to more dynamic information. Contracts are meant to be stable, but updating will of course occur. The writers were afraid that the technology would force an evo-

lution from catalog-as-contract to catalog-as-encyclopedia. In this way the meaning of the catalog would change, and it would no longer be a contract, but more a source of information. However, updates occurred as notices on boards. The students are committed to paying attention to the boards. The catalog, like documents in general, has both fixed and fluid properties, independent of which media they are based on (Levy 1994).

We regard it as important to organize updates in the catalog in such a way that the catalog is still regarded as stable and at the same time a source of relevant information. This is not necessarily a contradiction. Information about deadlines, intake, rules, and so on, is relatively stable, and it has to be. Information about course times and locations is more dynamic, and the students know that this information may change during the semester. To make it possible to rely on the electronic catalog, conventions such as ‘keep an eye on the notice board’ in the ‘paper world’ have to be developed and introduced in the ‘electronic world’ as well. Probably, these conventions will be developed over time (Brown & Duguid 1994, Yates & Orlikowski 1992). However, we propose to develop new ways to take this convention into account, and make them explicit for the organization.

We observed that the catalog had underlying and “hidden” intentions, which are developed over time and integrated into ongoing practice. The catalog is more than a medium to communicate information from the administration to students; it is an artifact that also coordinates work practice, e.g. the writers’ coordination of the updates and the students’ organization of the semester.

To produce the catalog for the next semester, the writers take the previous year's catalog as a starting point and update dates, delete old courses, and so on before they print out their pages. However, there are blank spaces and question marks in the text, since this is just a draft. The printouts are sent to the lecturers and others that have opinions and the right to make changes. The lecturers make their updates on the paper, and return it to the writer. The writer coordinates the rooms and types the new and updated text into the catalog. The 'new' catalog is distributed in the same way for proofreading until it is finalized.

The main goal of the catalog is to be a tool for students to plan and organize their semester. They use the catalog to get to know about the studies at the University, to decide which courses to take part in, what time they want to have the group lectures in order to avoid conflict with other courses or part-time jobs, and so on.

These two examples show that the catalog is integrated into ongoing practice, and not something 'added on'. There is no limit on how the catalog can be used. We have given only two examples here. The standardization of the catalog impacts the catalog in use, and it is important that the local need for flexibility of use is regarded in design. We have to be aware that the catalog, or other documents, is embedded in practice, and has roles there. This is further discussed in Braa and Sandahl (1998b). As an example, at least the printing facility, as discussed in the section above, has to function properly to support the need for drafts. In addition, the electronic version of the catalog should help students to

plan their semester, because that is one of the key objectives of the catalog.

## 8. Concluding Remarks

This paper describes the first steps of designing an SGML-based infrastructure. The goals of the pilot project presented were to produce a better catalog through a better structure and layout, to make it easier to update and maintain, and to gain practical experience in SGML. We also wanted to develop services like room allocation, advanced search and customized publishing based on the structured catalog. However, in the period of the pilot project we did not achieve all these goals. The design and development had a technical focus, and significant problems related to use occurred as a consequence.

Based on our experiences in this pilot project, we have the following concluding remarks:

1. In order to benefit from SGML-based documents, they have to be produced in the first place, which requires a satisfactory writing environment. Selection of the editor or word processor to be applied is essential, and has to be taken into account in DTD design.
2. Documents are integrated into work practices; they have roles, and conventions grow around them. An SGML implementation may (differentially) change these roles and conventions and thereby impact work practices.

These points are discussed in more detail below.

### 8.1. DTD Design

The DTD has to be designed in a way that supports the functionality desired. From a technical perspective a strict and rich DTD is preferable. From a use perspective the DTD should be flexible and as small as possible. We have indicated three different approaches to the problem. First, the DTD can be made more flexible by reducing elements and changing required sequences to optional sequences where appropriate. This is the easiest way to reduce strictness, and the most obvious. Second, the DTD should be made smaller by reducing the number of elements. This can affect the functionality, because some definitions disappear. However, in our experience there were far too many elements from the start, and the reduction of elements did not necessarily affect the functionality desired. Third, 'writers' DTDs' can be introduced as a subset of the full DTD. These DTDs consist only of elements that a particular writer uses. The effort of maintenance has to be taken into account before implementing a solution like this. The same goal is reached if each writer sees only the elements (s)he needs. The DTD is the same, but the editor 'hides' the unnecessary elements. Using predefined forms for each writer is an equivalent solution. In short, the DTD design impacts both the writing and the presentation or functionality, but in opposite ways. What one should emphasize depends on the situation at hand (competence of writers, degree of functionality, and so on). We state that SGML editors are better tools for producing text based on a complex DTD than WYSIWYG word processors are. On the other hand, if the DTD is simple and small, a WYSIWYG word processor should be applied.

The fact is that the WYSIWYG word processors are highly integrated into work practice. The writers know them and use them daily. However, if there is a need for continuous conversions to and from SGML to provide continuous updates, the use of templates can be a difficult approach, because of the errors that often occur in conversions (Maler & Andaloussi 1995).

### 8.2. Catalog in use

Because of the paper documents' ecological flexibility, they easily fit into different situations for staff and students. We have observed conventions related to updates of the paper catalog, conventions among coordinating the updates, and students' use in the organizing of their semester. The catalog is essential in these situations; however, it plays different roles. To be able to achieve a critical mass of use (Grudin 1994) the electronic catalog has to be flexible enough to be adapted to different situations. Some conventions related to the paper disappear, and new ones have to be introduced to get the electronic documents adopted into the organization.

### 8.3. Goals That Have Been Accomplished

In a survey carried out by a master's degree student, students reported that they were satisfied with the 'new' printed catalog. It was experienced as more structured than the earlier ones, and easy to navigate in (Markussen 1998). The accounting department reports that the catalog costs 200,000 less than the earlier versions not based on SGML. The reason is that the number of pages has decreased (Ibid.).

#### 8.4. *The Pilot Projects' Implications for Further Development*

Further development based on this experience has been carried out to achieve more of the overall goals. In brief, we can mention:

##### 8.4.1. *Catalog as One of Several Documents*

The organization has gained a whole new understanding of the content of the information produced, and there are several initiatives from different units at the University to deal with different document types according to the solutions for the catalog. Several document types in addition to the catalog now constitute an infrastructure of information, making it possible to reuse information components across different types of documents. Future work will address further integration with other information systems, e.g. database solutions for student systems for all universities and colleges in Norway. This has the potential to improve students' possibilities for planning the semester and signing up for exams.

##### 8.4.2. *DTD Design*

Modular DTDs have been developed to tailor DTDs to specific document types and to reuse general structures of elements between the DTDs.

##### 8.4.3. *Functionality for the Writer*

Administration routines are available through the WWW, including basic functionality such as publishing a document (making it available among the official WWW information), archiving and restoring data, ordering printouts and conversion to the local WWW (to look at the result before publishing). Functionality is tailored to suit different document

types, e.g. extracts of the information on lecturers and lecture rooms in the catalog for support of the planning work.

##### 8.4.4. *The WWW Presentation*

The conversion routines have been developed further to provide integrated solutions for including the university's visual profile on every HTML page, including relevant meta information such as "last updated by", who is responsible and contact points. In addition, there are better navigation tools to show the relevant context of the HTML page at all times, and improved search facilities. It is still a challenge to tailor the WWW presentation to different user groups, e.g. new students, researchers, or administrative staff.

##### 8.4.5. *The printing*

Much work has been done to improve the printing facilities. Today, the SGML files can be printed regardless of their status.

##### 8.4.6. *Continuous updates*

The organization has developed strategies and routines for dealing with continuous updates for some of the document types, e.g. the catalog. The aim of these strategies is to make it clear how late the writers can make updates in relation to deadlines for printing, and how and where students can catch up on updates in the electronic version. The latter has been designed, but not yet implemented at all levels at the university.

---

### **Acknowledgements**

Sincere thanks to Kristin Braa, Frieder Nake, Pamela Gennusa, Jonathan Grudin, Eric Monteiro, and Pål Sørgaard, as

well as the members of the SGML group at the University of Oslo: Bjørn Ness, Erlend Øverby, and Håvard Fosseng, for constructive comments on drafts of this paper. We would also like to thank Atle Holmer Markussen for giving us valuable new input to this paper as part of his Master's research on the catalog case. This work was supported by the Research Council of Norway through its grant to the BEST program and the Swedish Transport & Communications Research Board through its grant to the Internet Project.

## References

- Alschuler, L., (1995). *ABCD ... SGML A user's guide to structured information*, International Thompson Computer Press.
- Berry, M. D. and M. A. Goulde, (1994). A New View of Documents. Integrated Information Management in the '90s. In *Workgroup Computing Report*, 17(8).
- Braa, K. & T. I. Sandahl, (1998). Approaches to Standardization of Documents. In T. Wakayama, S. Kannapan, C. Meng Khoong, S. Navathe & J. Yates, editors. *Information and Process Integration in Enterprises. Rethinking Documents*. Kluwer Academic Publishers.
- Braa, K. & T. I. Sandahl, (1998b). From Paperwork to Network. *Third International Conference on the Design of Cooperative Systems (Coop '98)*, Cannes, France, 26.-29. May 1998.
- Brown, J. S. & P. Duguid, (1994). Borderline issues: Social and Material Aspects of Design. In *Human Computer Interaction*. 9:3-36.
- Date, C. J. (1986) *An introduction to database systems*. Addison-Wesley Publishing Company.
- Goldfarb, C. F., (1992). *The SGML Handbook*. Clarendon Press, Oxford.
- Grudin, J., (1994). Groupware and Social Dynamics: Eight Challenges for Developers. *Communication of the ACM*, 37(1):93-105.
- Hanseth, O., E. Monteiro & M. Hatling, (1996). Developing information infrastructure: the tension between standardisation and flexibility. *Science, Technology and Human Values*, 21(4):407-426, Sage Periodicals Press 1996
- van Herwijnen, E., (1993) *Practical SGML*. Kluwer Academic Publishers.
- Jenssen, A. E. & T. I. Sandahl, (1996). Conflicts between the possibilities and the reality in the field of structured electronic documents: Experiences from a large-scale SGML-project. In: B. Dahlbom *et al.*, editors. *Proceedings of the 19th Information Systems Research Seminar In Scandinavia*, pages 935-955, Lökeberg (Göteborg), 10-13 August 1996.
- Jewett, T. & R. King, (1991). The Dynamics of Computerization in a Social Science Research Team: A Case study of Infrastructures, Strategies and Skills. *Social Science Computer Review* 9(2). Duke University Press.
- Knuth, D., (1984). *The TeX-book*, Addison-Wesley, Reading, Massachusetts.
- Lamport, L., (1986). *LaTeX, user's guide and reference manual*, Addison-Wesley Publishing Company, Reading, Massachusetts.
- Levy, D. M. (1994). Fixed or fluid? Document stability and new media. In: *Proceedings of ECHT'94*. ACM, pp 24-31, New York.
- Maler, E. & J. E. Andaloussi, (1995). *Developing SGML DTDs: From text to model to markup*. Prentice Hall.
- Markussen, A. H., (1998). *Fra produkt til tjeneste i dokumentbaserte IS. Perspektiver og utfordringer*. Master Thesis (in Norwegian). Department of Informatics, University of Oslo.
- Reinhardt, A., (1994). Managing the New Documents. *Byte*, 19(8).

- Smith, J. M., (1992). *SGML and related Standards. Document description and processing language*. Ellis Horwood Limited.
- Star, S. L. & K. Ruhleder, (1994). Steps Towards an Ecology of Infrastructure: Complex Problems in Design and Access for Large-Scale Collaborative Systems. In: *Proceedings of CSCW 1994*, ACM. Chapel Hill, NC, USA, pp 253-264.
- Sørgaard, P. & T. I. Sandahl, (1997). Problems with Styles in Word Processing: A Weak Foundation for Electronic Publishing with SGML. In *Proceedings from 30th HICSS*, Wailea, Hawaii, Jan 7-10.
- Ventura, C. A., (1988). Why Switch from Paper to Electronic Manuals? In *Proceedings from Santa Fe*, New Mexico, December 5-9.
- Yates, J. & W. Orlikowski, (1992). Genres of Organizational Communication: A Structural Approach to Studying Communication and Media. *Academy of Management Review*. 17(2):299-326.