Master thesis

Information exchange for the book industry in Norway

Written by
Reidun Husby

Department of Informatics
University of Oslo
Norway

April 2005

Abstract

The book industry in Norway consists of many different types of companies. We have the publisher who publishes the book, the Distribution Central who distributes the book from the publisher to the book store and the book store who sell the book to the customer. There are also several other company types like the National Library, the Norwegian Library Bureau and the Arts Council Norway that belong to the book industry, but in this thesis I will mainly focus on how the exchange of book information is between the publisher, Distribution Central and the book store.

In order to learn how the information exchange is today and uncover problem areas with this solution, I interviewed seven companies from different parts of the book industry. Several problem areas were uncovered from the interviews.

One of the problem areas was concerning the lack of standard when the publisher send book information to the book store through the Distribution Central or any other company that requests book information. Since there are agreements between the sender and receiver how the information should be sent and that company use various format and structure, makes the information exchange ineffective and time consuming.

I will therefore explore whether the book industry should adopt a standard format for exchanging information and how this could be done. There exist today an international standard for information exchange. I will therefore look into how this standard works, and how the book information sent using this standard could be stored in a database.

Preface

This master thesis was started in January 2004, and is a part of my Master Degree at the Informations Systems group at the Department of Informatics, University of Oslo.

Tone Bratteteig from the Informations Systems group and Peter Hausken from J. W. Cappelens forlag have been my teaching supervisors. Tone is the leader of the Information Systems group, and Peter has been working as the Chief Information Officer (CIO) at Cappelen since 1995.

During the last year and a half I have been exploring how members of the Norwegian book industry exchange book information. I have interviewed companies from different parts of the book industry to see what their needs are, and if their needs are fulfilled today.

Many big thanks go to Peter Hausken for his enthusiasm, support and guidance through the whole period. You have been a great help!

A big thank you also goes to Tone Bratteteig for you guidance and advice regarding this thesis.

I also want to thank the publishing company, the two Distribution Centrals, the three book stores and the National Library for taking the time to meet me, and giving me important information for my research.

Table of content

Figures

Tables

Chapter 1
Information exchange in the book industry

Picture you walking into a book store, or browsing on the web for a book. Do you normally know what kind of book you are looking for, or are your choice based upon what kind of information a book contains? If you know which book you are looking for, how did you receive this information?

Information about a book is important in order to inform a potential buyer about a book, so he can decide if he wants to buy it or not. Information valuable for the user could be everything from the name of the author, title of the book or what kind of book it is.

The book information is originally designed by the author of the book. In order to get this information from the author to the buyer of the book, several companies within the book industry are involved. The publisher stores the book information designed by the author along with some firm specific information like price, weight and publishing date. The publisher then sends the book information to their central, book stores and libraries, to inform you and me about the book. There are used different format for sending this information, depending on which company that shall receive the book information.

In this thesis I will investigate how companies in the book industry in Norway exchange book information, and their thoughts around how this is today. I will also look at what kind of information need the book stores have in order to provide the customers with enough information about a book. This will be interesting to look into, since it is a rather long process to get the information about a book from the publisher and author, to the book store and the potential buyer. On its way, the book information gets edited, classified, and presented several places. The information exchange process is time consuming since there is not used any standard format when exchanging the information. I will therefore look into changes that can be done with the information exchange solution that exist today, in order to make it more effective.

The objective of this thesis is to determine what kind of information about books needs to be exchanged between publisher and book store, and how this is could be supported by IT.

I will start this chapter by giving a description to some of the companies in the book industry. I will then look at how the information exchange is done in the book industry today. At the end of this chapter I will inform about the method used to collect information for my research, and how the structure of the rest of the thesis will be.

1.1 About the book industry

The book industry in Norway consists of the following company types:

- Authors
- Publishers
- Preprint agencies
- Printing agencies
- Distribution Centrals
- Book stores
- Libraries

In this thesis I will mainly look at how the publishers, Distribution Centrals and book stores exchange book information with each other, since how the information gets sent from the publisher to the book stores are the most important part of the information exchange process. Other companies like the Norwegian Library Bureau[1], the Arts Council Norway and press also receive book information from the publishers, but this will not be focused on in this thesis. Besides looking at how these three company types exchange information, I will also look into which part the National Library (NL) play in the information exchange process today.

The figure below illustrates the relationship between the three company types:

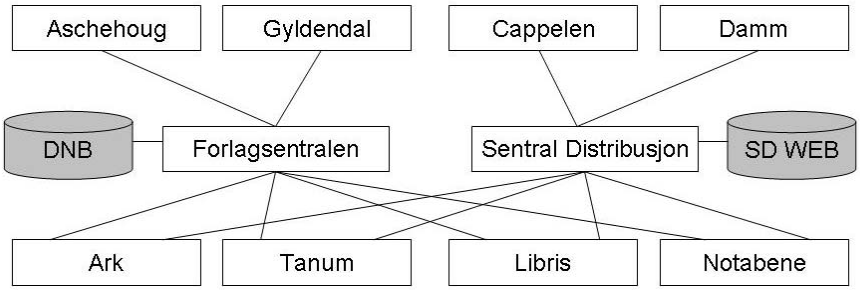

Figure 1.1: How the publisher, Distribution Central and the book store are connected

Figure 1.1 illustrates how four major publishers are connected to the two Distribution Centrals in Norway, and that the two Distribution Centrals are connected to all the book stores within the book store chains. I have placed a database with the name DNB, short for the Norwegian book database, next too Forlagsentralen (FS). This database contains information about most books published in Norway. Next to SentralDistribusjon (SD) is a database with the name SD WEB. This database contains information about books published by the publishers connected to SD. The book stores are connected to both the two Distribution Centrals, and can order books from both. The book stores above are really book store chains, since each of them consist of many stores. Notabene for example is a book store chain which has somewhere around 100 stores located in Norway.

There will now be a description of the three company types, before a general description of how the information exchange done today.

1.1.1 Publisher

The publisher's job is to read through a writer's material, in order to decide if they want to publish the book or not. If the publisher decides to publish it, they have to correct errors and prepare the book for printing in cooperation with the author. The marketing of the book is also done by the publisher, which means that the publisher has to send the book information about the book to many organizations. The book information is sent to the Distribution Central the publisher is using. When the book is ready for printing, the book's text is sent to the printing agencies. The Distribution Central receives the books when they have been printed.

For schools, the situation is different. Schools receive a suggested list of titles from the publisher and it is only those books, chosen by the schools that are published.

The Norwegian Publishing Association (DNF) had forty four publishers as members on the 1st January 2004. These members had 75-80 % of the total turnover for books in 2002 together with their book clubs. The four publishing companies from figure 1.1; Aschehoug, Gyldendal, Cappelen and Damm, are together with Schibsted-Forlagene the five largest Norwegian

[1] The Norwegian Library Bureau is in charge of purchase for the Norwegian culture counsel, and sells publications to libraries

publishing companies. (DNF 2004)

A publisher is use one of the Distribution Central, but there are also some small publishers that only publish a book or two, and are therefore not connected to any of them.

During the last 40 years have there existed almost 50 book clubs in Norway. In 2002 the Norwegian book clubs had a 25 % turnover of the total book market. The book clubs can offer the books with a discount right after they are published (DNF 2004).

The book club market is dominated in three groups (DNF 2004):

- The Norwegian book clubs, owned by Aschehoug, Gyldendal and Pax
- Cappelen's book clubs
- Damm's book clubs

1.1.2 Distribution Central

There exist two Distribution Centrals in Norway:

- SentralDistribusjon (SD)
- Forlagsentralen (FS)

In the early 90's, FS was a great deal bigger then SD (Solberg 2004), but this changed somewhere around 2000 when several large publishers moved from FS to SD. At the end of 2004 SD had a 47 % marketing share according to the marketing director at SD (SentralDistribusjon 2005).

He also told me why he thinks several publishers moved from FS to SD:

1. SD and FS used different models towards their publisher, which made SD more cost effective than FS.
2. The two publishing companies who own FS also own book stores. This is not an ideal situation because of the possibilities of conflicting interests.
3. SD has better customer relationship with higher service level, greater flexibility and simplicity.

According to numbers given about the two Distribution Centrals (DNF 2004):

- SD has 70 publishers as customers, and distribute and store around 19900 titles
- FS has almost 300 publishers as customers, and distribute and store around 20500 titles

The Distribution Centrals job is to store and distribute books from publishers to book stores (Forlagsentralen2).This includes taking care of the book from when it is printed until it is in the book store. In other words: receiving the product, storage, setting up orders, transport, return, shipping free copies and copies that have to be sent to NL.

FS and SD are both owned by publishing companies. FS is owned by Gyldendal and Aschehoug with a 50 % equal share, while SD is owned by Cappelen and Svensk Film, where Cappelen own 99 % and Svensk Film 1% (DNF 2004).

There are three places to search for information about books. Both Distribution Centrals provide the users with a register where a small set of information about their publishers books are stored. The marketing director at SD told me that the users can also order books and look at orders placed earlier from SD web. These registers are free to use. Besides these two registers, FS provides users with access to Mentor, which contains all the information stored in the DNB database, only stored a bit different to increase the speed of searching. It costs money to use Mentor, but is much more advanced than the two other registers. Mentor is the database that book stores, publishers and libraries use to look up information about books, and book stores can order books from. I will give a more detailed description of Mentor in the next section. The background for that FS has the

responsibility of DNB is of historical reasons according to an IT manager from the book industry.

Another thing FS does is to deliver book information to book stores, so they can store it in their system.

1.1.3 Book store

Today there are somewhere around 600 book stores in Norway (DNF 2004). The book stores are not obligated to any of the Distribution Centrals, but can place orders to both. Stock inventory is used by many book stores to keep track of information on how many books they have in stock and how many books they have sold. Stores who use stock inventory order books directly from their own system, and not from Mentor or SD web. When orders of books are placed, the order goes directly to the Distribution Central that the books publisher uses. For stores who do not have a stock inventory, they can order directly from Mentor or SD web. Orders can also be placed by phone, fax or even letter.

Around 60 % of the total turnover of books is done by book stores. Around 90 % of this the turnover made by the book stores are done by the following book store connected to a chain or other cooperation: ARK, Norli, Libris, Interbok, Notabene, Unipa and Fri Bokhandel. Libris and Interbok make BKN (Bok og kontorkjedene I Norge) (DNF 2004).

Several book stores sell their books also on web. The number of customers that buy books on the web is increasing. According to MMI 5 % of the Norwegian population bought one or more books through a Norwegian web store in 2002, and 3 % through a foreign web store. One of the most important reasons for buying books on the web is that you get access to books you normally will not find in the shops (DNF 2004).

1.2 Mentor

Mentor and DNB are databases with information about books controlled and maintained by FS. The book information is added in DNB by FS, and copied over to Mentor in order to make it easier for the users to search for. Mentor contains information about 99.5% of all books published in Norway. The books do not have to be available to be able to find them in Mentor, but can be sold out or not yet printed. (Forlagsentralen1)

Below is a figure given by FS, which illustrate how the two databases are connected and which role, FS, the book store, the librarians and the publishers have.

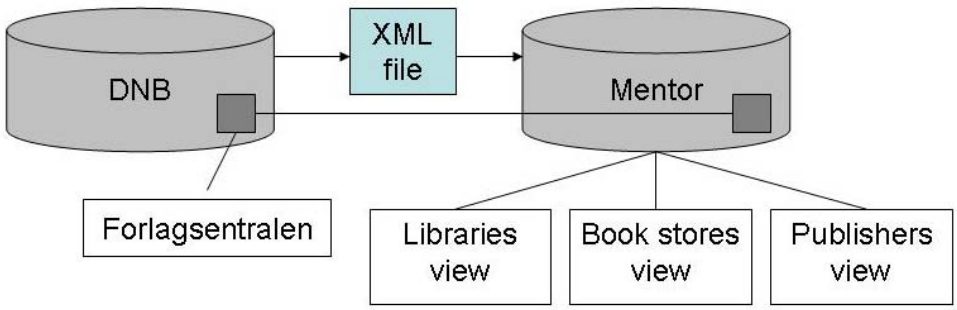

Figure 1.2: Connection between DNB and Mentor

Both the DNB and Mentor database contain information about books published in Norway, whereas Mentor contains much of the information stored in DNB. Mentor is built a little bit different to make searching fast. The views connected to Mentor are the views the libraries, book stores and publishers can see. The little square inside the Mentor database is where the

information sent by the publisher is stored and the one inside DNB are where the staff at FS add/edit the information.

A book does not have to be added in DNB, but most publisher want to add their book in DNB, so book stores can find it in Mentor. In order to get the book information in DNB, it has to be sent to FS physically.

Today, somewhere around 90% of all book stores are connected to Mentor (Forlagsentralen1). Some of the things Mentor can be used for include; searching for books using specific criteria, ordering a book or looking at orders placed earlier.

It is not only the book stores that use Mentor. Publishers use Mentor if they want to update some information about a book like price and status information, and to see which books that is on the market. Mentor is also used by libraries.

When a publisher wants to add a new book, they send the information about it to FS by either using the publisher view, mailing it or a file containing it, or even by regular mail. When the publishers (connected to FS) or SD (send on behalf on their publishers) sends the information to FS, this is done using all kinds of format. The information SD sends is added to FS' system, before it is added into DNB. If the information is added through the publisher view, the information will go to the receiver base, since the publisher can not add the information directly into DNB. Typing in the book information using the publisher view is a time consuming job, and is therefore usually not done for larger publishing companies. For these companies it is much easier to send the information electronically, since the information is already in their systems.

FS gets the information from the receiver base in Mentor to their receiver base in DNB. It is here all the work with the information is done. The information FS gets from the publishers is usually only a small set of book information like ISBN, title and author. Classifications, book description and pictures of the book are information added by FS, while missing information from the publishers is also found. The book information is also quality checked before it is added in DNB. It is therefore important that FS get a copy of the book itself to find the additional information and check it. Once an hour updates are sent from DNB to Mentor using XML. Adding a book to DNB is a long process, since FS have to type additional information about the book.

After the book information has been added in DNB, the book stores, libraries and the book's publisher can access the information through different views in Mentor. The book stores and libraries can choose between a simple and an advanced version of viewing book information, whereas a simple version only shows a register with information about Title, Author and ISBN for books FS have in stock. The information in this register has not been quality checked. The advanced version gives access to Mentor, where they can view much book information independently of where the books publisher is connected. I was told that the register is not a simple version of Mentor, since the information in this register comes from their logistic system. There is a small amount of interest in this register among the book stores and libraries. About 550 book store subscriptions and 200-250 library subscribes Mentor today. Some libraries get access to Mentor through special agreements. The simple version is similar to what SD offers their customers. The difference in cost is that both the SD and FS registers are free while the use of Mentor costs money.

I have included a picture below which show what the book stores see of information about a book using Mentor.

Figure 1.3: Information about a book in Mentor

Publishers can choose between having a limited version of Mentor, or the full version. If they choose to have limited access they get access to their own titles, and can register books to FS without paying anything. Information like price and status often changes, and the publishers can therefore use Mentor to update this information. There are also some publishers who choose to have access to the full version of Mentor.

Besides providing much information about books, Mentor also provides access to a couple of foreign databases, where users can find information about books not published in Norway.

1.3 The information exchange today

I will here give a description of how the companies in the industry exchange information about books, based upon information from the interviews done in the following chapters and what an IT manager from the book industry told me.

The information about a book that is going to be published is stored in the publishers system. The marketing of a book is done by the publishers, which mean that the publishers have to send book information electronically or published in magazines to companies with interest of this information. Examples of companies that publishers send information about books to are their Distribution Central, book stores, the Norwegian Library Bureau, the Arts Council Norway and the press.

The information sent to their Distribution Central can be sent using a web registration, mail, post, or directly export of information from the publishers system to the Distributions system. SD has to send the book information for their publishers to FS, in order to get the book information stored in DNB and visible in Mentor. This is because FS is the one who add this information to DNB.

FS receive the book information in various formats. This is because the information is sent

between the sender and receiver using agreements on how the information should be structured so the receiver knows what the information means.

The staff at FS type into DNB the book information they receive, after the information has been classified, edited and quality checked. Since there is no standard format when sending the information to FS, they can not get the information added to DNB directly. The information comes in various formats and has different structure. This leads to a lot of extra work since both the publisher and the Distribution Central has to type in the book information to their systems.

Besides adding book information into DNB, FS also sends book information to many book stores so they can add it to their systems. There is not used a standard format when sending this information either. The book stores I talked to add the information sent from FS directly to their systems. When the book stores shall look up information about a book, they use their own system or Mentor. Mentor is subscribed by 550 of book stores (around 90%), and from the interviews I were told that this is because Mentor is the only place where information about all books published in Norway is stored. If a customer comes and asks for a book, they will always find it in Mentor.

I have not explored how the book information is received at the Norwegian Library Bureau, the Arts Council Norway, the press and other companies that may receive book information from the publishers, but focused on how this is done at the Distribution Central and the book stores. However, after talking to a person at the National Library (NL) I was told that they do not receive any book information from the publishers at all. They have to find the book information themselves, and add it to their systems.

1.4 ISBN

The International Standard Book Number (ISBN) has been an international identification system for the publishing industry and the book trade since it was introduced in 1970. (ISBN 2004) The 10 digit ISBN identifies the book. This means that two books can not have the same ISBN. The ISBN is not any random number, but some of the digits identify the country and the publishing company.

Looking at the ISBN 8202231868, the digits 82 is a reference to Norway, 02 for Cappelen, 23186 is the book number, and the last digit is a check digit. Not all countries have this type of division of the ISBN, but might have the first four numbers for example as a country code.

The guidelines from the International ISBN Agency (ISBN 2004) describe how the ISBN is about to change. Because of the growing number of publishers and publications around the world, the amount of ISBN is beginning to run out. The ISBN therefore needs to be expanded. The new ISBN will have 13 digits, and will be identical to the European Article Number (EAN) that today appears together with the ISBN on the backside of the book. Expanding the ISBN to 13 digits is the first change that has been done to the ISBN since it was introduced in 1970. The new ISBN has to be implemented by 01.01.2007. This gives the book industry time to adjust their systems to the new ISBN, and discuss how and when this should be done.

The first 3 digits in the new ISBN will identify the book industry, the next 9 digits are the core number, and at the end a check digit.

A document given by the NL (Nasjonalbiblioteket 2003) contains information about what can be done when the new ISBN is adopted. It is recommended that companies should deliver book information to all ISBN offices, and that the ISBN offices shall create and maintain databases. In Norway NL is the Norwegian ISBN office. They provide the publishers with the ISBN. In order to start receiving book information from the publishers, the document mentions that NL could say

that they need to receive the information before the publishers would get the ISBN. More about this will come in chapter 4, where I interview someone from NL.

1.5 Method used

I want to uncover problem areas with today's way of exchanging information in this thesis. To do this, I am going to interview companies from different parts of the book industry to learn about how they exchange information with others, and problem areas they see in today's solutions. Since the book industry consists of several companies and organizations, I will not be able to look into how all of them are exchanging information with the rest of the book industry, and thoughts around this. I will focus on how the publishers exchange book information with the book stores through the Distribution Centrals, since the most important part of the information exchange is to get the information to the book stores and the potential buyer.

Besides not looking into how all companies and organizations in the book industry exchange information with each other, I will not look at all the problem areas uncovered in the interviews. I will focus on what kind of information that needs to be sent from the publisher, and how this could be done in order to make the information exchange solution more efficient. I will not look into problems concerning the ownership and maintaining of DNB, nor how Mentor could be improved, since these two problems are more related to how the book industry function, then how IT could improve the information exchange solution today.

Interview is a qualitative method mainly used to gather information.

Qualitative research methods are designed to help researchers understand people and the social and cultural contexts within which they live.

(Myers 1997)

Myers gave the following examples of different qualitative data sources: observations, interviews, questionnaires, documents and texts and the researcher's impressions and reactions. (Myers 1997)

Personally doing interviews helped me to get a deeper understanding of how the information exchange is really done in the book industry, instead of sending out questionnaires or only reading documents. Interviews are a time consuming method, because you will have to talk to every person interviewed (Valenzuela and Shrivastava). Personal interviews take even more time, since you will have to meet the person face to face. It would be very expensive to use this type of method for interviewing many persons.

Interview as a method has several advantages that you would not get by sending out questionnaires. You will be able to ask follow up questions, and it is easier for the person interviewed to come up with additional information not among the questions.

The general interview guide approach is an interview type that ensures that the same general areas of information are collected in all the interviews, but it also allows freedom of how the information should be collected (Valenzuela and Shrivastava). This type of approach was used in the interviews I did, since I wanted all interviewees to tell me about how the information exchange was done at their company today, and problem areas they felt were important today.

I also received a set of information from two book stores and NL, which they felt were important to have about a book. The information from the two book stores was not given during the interviews, but given at a later stage.

The following companies and organizations were interviewed:

- 1 publisher
- 2 Distribution Centrals
- 3 book store chains
- The National Library (NL)

The only interview group that got the same type of questions was the book stores chains, while the publisher, the Distribution Centrals and the National Library got questions specific to their organization.

The reason for the interviews was to collect information about how they exchange information with others in the book industry, and to see if there are any problems regarding how this is done. I also got a list from two of the book stores chains and the National Library, where information they found important was listed.

The three book store chains I chose to interview had according to statistics from 2003 marketing share of 33%, where the two chains that gave me the list over information they find important presented 27%. This number is based upon how many books they have sold, and not other things they sell in their store.

I have not mentioned any names of the people interviewed in the different organizations, since this is not relevant. The focus should be on what the interviewees said, and not who said what. When it comes to the Distribution Central, it is obvious that I interviewed FS and SD, since these is the only two Distribution Centrals in Norway. I have also written that I interviewed the NL, but not who from the NL I interviewed.

Besides interviewing parties in the book industry, there has also been e-mail correspondence with two organizations. I e-mailed two persons from EDItEUR[2], who are responsible for the maintenance of the ONIX standard, which is a standard for exchanging book information. I will come back to this standard in chapter 7. The reason for e-mailing them was to get their opinion on what kind of database solution they thought best supported the ONIX standard. I also e-mailed Amazon, an American book store. Amazon was among the first companies who adopted the ONIX standard. I wanted to know how far they had come with getting companies to exchange information using ONIX.

The last method used to gather information was to study articles where standards and databases are discussed.

[2] EDItEUR is the international group coordinating development of the standards infrastructure for electronic commerce in the book and serials industries EDItEUR3. Retrieved 02.03, 2005, from http://www.editeur.org/what.html.

1.6 The structure of this thesis

To get an overview of what this thesis is about, I will give a short introduction of the following chapters:

Chapters 2, 3 and 4 are from the interviews done.

Chapter 5 will contain a discussion of what can be done in the book industry based upon the problems discovered during the interviews. Suggested solutions to the challenges will be presented.

Chapter 6, 7 and 8 will be about standards in general, the ONIX standard in particular and the XML standard which is used in the ONIX standard. These chapters explain the importance of standards, and give an example of the ONIX standard. A description of XML and ONIX relation to XML will also be presented.

Chapter 9, 10 and 11 will explore several database solutions. Trying to find the one most appropriate for storing book information sent in ONIX format. I will also look at how an ONIX message with information about a book could be stored in the database. The last chapter will explore the possibilities of using the XML features in Oracle and SQL server.

Chapter 12 is the last chapter, and will summarize the findings done in this thesis. The book industry's possibilities based upon the findings through this thesis will also be explored.

Chapter 2
How two Distribution Centrals exchange information

In this chapter I will look at what thoughts Forlagsentralen (FS) and SentralDistribusjon (SD) have around the information exchange solution in the book industry today. Learning how the Distribution Centrals exchange information with publishers, book stores and each other, makes it easier to see if there is any room for improvements. The Distribution Central are the connection between the publisher and the book store, since they receive the information from the publishers, and one of them send it to the book stores, so any problem areas with this solution will be uncovered in these interviews.

2.1 The interview with Forlagsentralen (FS)

The interview I had with FS on the 17.02.2005 was with an IT consultant that has worked there for several years.

Besides talking about the ONIX standard, I asked the IT consultant the questions below. I wanted her to tell me how the information exchange with the other parties in the book industry was done. I did not get all my questions answered, but I got a good description on how the information exchange is today, and how it could be done in the future.

2.1.1 The questions

The questions I wanted FS to answer were within two areas:

- How the information exchange is done today
- How the information exchange could be done in the future

I hope to get their opinion on what their part of today's information exchange are, and how they can contribute to improve their role.

Questions regarding how the information exchange is done today:

1. What kind of companies uses Mentor?
2. How many companies of the total amount of companies in the book industry use Mentor?
3. How much do the companies pay to use Mentor?
4. How is the payment set? (type of company, per license, per company, amount of functions)
5. How much does it cost to maintain Mentor each year?
6. Does the payment from the companies cover all the cost?
7. Tell about how the information exchange is done in the book industry today, and which part FS have. (Add information into Mentor, manage order of books)
8. Can you make an illustration where you show me how the different systems are connected
9. Who sends book information to FS to be put into Mentor?
10. How many companies send book information?
11. Which format do they use?
12. Do you have to adjust these formats that the companies send the information a lot?
13. How many book records do you receive every week?
14. How much time do you spend on each record?

Questions regarding how the information exchange could be done in the future:

15. Are you happy with how the information exchange is done today, or do you see room for improvements?
16. What can be done to improve the information exchange?
17. How can the amount of time spent on each record go down?
18. Would it help if the publishers use a standard format like ONIX when sending you book information?
19. Could the publisher add all the information themselves, so FS only add the information in Mentor?
20. Could the publisher import the book information directly into Mentor?
21. How can the cost for the companies who use Mentor today go down?
22. Would it be possible to make it free for the book stores to use Mentor, and rather reduce the amount of information and functions? (Pay for extra if they need it)

2.1.2 The information exchange today at FS

I was given the illustration in figure 1.2 was given by the IT consultant, which show how the information exchange is done between the publisher, FS, book stores and librarians today.

Besides adding information about books in the DNB and Mentor as described in the section about Mentor in chapter 1, FS also send book information to book stores, so they can add it to their own systems. Most book stores receive book information from them. There is no common standard format for exchanging this information either.

There are several disadvantages with how things are done today. One of them is that it is very time consuming to type in the information to DNB. Each year there is around 10.000 new titles that shall be added to the book base, and before the information is added must it be edited and quality checked.

The interviewee told me that the book stores are happy with how Mentor works today. They think it is easy to find a title they are looking for, and they like the detail level of the information. There is much information about a book available. FS create a review of the book, soundtrack, connections between books and much more. This type of detailed information cost money to produce and is therefore dependent on demand and will to pay. FS is a commercial organization, and is therefore not interested in offering information and functions they do not get revenue from.

2.1.3 Standardizing the information exchange

A new ERP system and a new book base are today being designed by FS. They are looking at how the ONIX standard can be integrated with these two systems.

There are mainly two places they believe it can be valuable to implement the ONIX standard:

- When sending book information to book stores
- When SD send information about books on behalf of their publishers

Because FS send much information to book stores, it would be preferable to use a standard format. FS also receive a lot of information from SD, since SD send book information on behalf of their publishers. If the information would be sent using a standard format, it would be easier to import the information directly into their systems.

The interviewee does not think it is a good idea to implement the ONIX standard between DNB and Mentor. This is because an ONIX message contains many elements that they do not need. The XML file used to transport updates to Mentor today does not have many elements, but several attributes with information. By using a small XML file instead of a large ONIX message, IT will go very fast.

When it comes to how much time FS spend on each record, she does not think this would go down if the publishers start to send more information. She thinks the publishers instead should focus on the information they send being correct. FS quality checks the book information the publishers send over anyway, so there is no point for the publishers to spend time on trying to classify books. There are several people who work with this at FS. It is therefore easier to find the right groups for a book here; since they can discuss which group a book belongs to.

2.2 The interview with SentralDistribusjon (SD)

I interviewed the Marketing Director from SD on the 24.02.05, who has worked there since 1997. By interviewing SD I wanted to see how they feel about today's information exchange solution, and if they have any suggestions on how today's solution can be improved.

Besides getting my questions answered, the marketing director told me a great deal about SD, and its history. Marketing figures show that SD has grown tremendously the last couple of years. From having a marketing share around 20 % in 1995, they have today almost 50 %.

2.2.1 The questions

I had prepared the following questions:

1. Who does SD exchange information with today?
2. How do the publishers send their book information to SD?
3. What is done with the information sent by the publishers?
4. Tell about how the information exchange is done between SD and other parties today?
5. How can a standard improve today's way of exchanging information?

2.2.2 Information exchange today at SD

The marketing director told me that the following things are done when their publishers are going to publish a new book:

The publisher sends the printing agencies the book's text. Earlier or at the same time they send the book information to SD using SD web (SD). SD has a form where the book information has to be typed in. Not all publishers type in the book information using SD web. For a small publishing company, SD helps them filling in the information. There are also one or two publishing companies who send the information electronically to SD, and are then added directly into their system.

After the publisher has given SD the book information, the information will automatically be transferred to FS, where it will be added to DNB. When SD receives the printed books from the printing agencies, they inform the publisher that the book is ready for sale. If the book is printed for the first time, there are several things SD has to do like measure the book and send copies of the book to NL.

Book stores can access SD web to read information about books and to order books. Ordering books using SD web is not much used today because orders are usually placed directly from the book stores system and from there transferred directly to the Distribution Central. From SD web the book stores usually only read the status of the book and the status of the orders placed by the book store. Orders are placed directly to SD. There is however some book stores who place their orders using Mentor. Around 5-6% of the total turnover on books is made by book stores ordering book using Mentor. SD has to check Mentor each day for orders. SD has to pay for checking Mentor for orders. If there are any orders placed to SD, SD will have to collect this, and send out books to the book stores.

In his opinion the problem with today's way of exchanging information are around DNB and Mentor maintained by FS. He feels that all parties in the book industry pay too much money to

have FS classify and maintain the book information in DNB and Mentor. It uses too many resources to have staff at FS classify the book, and type in the book information again, when the publisher has already typed in the information once.

2.2.3 What can be done with the Information exchange

I was told that SD and their publishers are happy with how the information exchange between the publisher and them are today. However this process can be improved if a standard format for sending information is adopted. It would than be easier for the publishers to send the book information directly to SD instead of typing in the information at SD web. For some publishers it will still be best to send SD the book information using SD web, since they do not have the appropriate local systems or the knowledge of how to update their systems.

Implementing a standard would also make it easy to extract information from SD's system, and get the same structure on the book information. Below I will illustrate how the information exchange between the publisher and SD would look like if they used a standard.

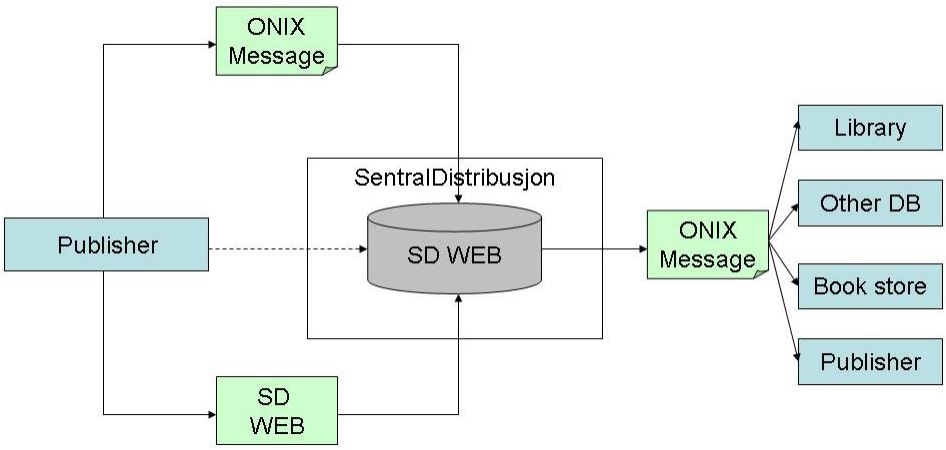

Figure 3.1: Information exchange between SD and their publishers

Figure 3.1 illustrate 3 forms of adding information about books:

- Adding the information using SD WEB
- Direct communication between the publishers system and SD system. Only one publisher has this today
- Adding information by sending SD the information using a standard format. The information in the format can be extracted from their system, and added in SD's system.

From figure 3.1 you can see that the information stored in SD's system could be extracted by help of a standard. The information extracted can be sent to for example book stores, library or other DB. In theory the information can be added in DNB directly. The publisher can add all the information, so the staff at FS does not have to classify and edit and add the information.

2.3 Summary of the interviews

The two Distribution Centrals had much the same opinion regarding how the information exchange solution was working today, but their solution to these problems differed some.

The Distribution Centrals told me that it takes much time to get a book added in DNB. After a publisher has sent over his book information to his Distribution Central and FS has received this from SD if the publisher was connected there, FS do not add the information directly, but go through all the information to see if it is correct, add classifications and additional missing information, add a book description and scan the front picture of the book, before typing in the

information to DNB.

Both Distribution Centrals feels that a standard format would improve the information exchange solution they have today. By using a standard the Distribution Centrals will receive the information from the publisher using a standard format, the information exchange between SD and FS becomes easier and FS will be able to send the information using a standard format to the book stores.

However, SD feels that a standard opens for adding the book information directly into DNB and that FS do not have to quality check and edit the information. FS on the other hand do not think that the information should be placed directly into DNB; before they have quality checked it and added additional information.

Chapter 3
How tree book store chains exchange information

In this chapter three book store chains told me about the systems they use to get information about books and to order books. By interviewing three chains, I got a deeper understanding of what kind of systems they use and how they communicate with others in the book industry. I want to uncover what the book stores are satisfied with, and what they want changed.

I wanted the three book store chains to give me an answer to the following three questions:

- What kind of system they use when exchanging information with others in the book industry
- How they find today's information exchange solution
- What kind of needs they have of a system

Based upon the three questions above, I designed nine questions that I was going to ask the interviewees:

1. Who in the book industry are you exchanging book information with and how is this done?
2. What is the book information used for? Do you have to do anything with the information before you use it?
3. Which systems are used to get the book information and to order books?
4. What kind of book information do you need from Mentor? Does Mentor fulfill your information needs?
5. What kind of functions do you use in Mentor? Does Mentor have all the functions you need?

If they use systems besides Mentor:

6. Why do you use systems besides Mentor? Does not Mentor cover all your needs?
7. What are the differences between the systems? Is there something that makes the other systems better then Mentor?
8. Is it something you are not satisfied with when it comes to Mentor or other systems used?
9. Are you satisfied with how the information exchange is today? What do you think can be done differently? What would a system that covered all your needs look like?

3.1 The interview with book store chain 1

Book store chain 1 is a small chain when it comes to the number of stores, but not when it comes to the results. The chain has less then 10 stores connected to them with central locations. In 2003 had the chain the best drifts result.

I interviewed the IT-manager from book store chain 1 on the 02.07.04. She explained to me how they communicate with others in the book industry, and she described the systems they use to exchange information with them. She also showed me how they order books, find information about books and information about orders placed earlier using their own system and using Mentor.

3.1.1 The information exchange today

Stock inventory is used to store the book information, and keep track of the books they have in stock. For books published in Norway, they use Mentor to get hold of the book information. The book information found in Mentor is put into their own system. Besides the information stored in

Mentor, they have to fill in some company specific information to their system. The information requirements are covered by Mentor.

Besides Norwegian books, they sell a lot of foreign books in their stores. The information about these books can not be found in Mentor, since they order these books from different countries websites. The book information on foreign books is received on paper, which means that they have to type it in to their stock inventory. They are working on improving this, since this is very time consuming.

A couple of times a week book store chain 1 go through the sales statistics in the stock inventory, to see which books they have sold over the last period, and how many they have left. If they want to place an order for some more books they do this from the stock inventory. The order will go to either Forlagsentralen (FS) or SentralDistribusjon (SD), depending on the books publisher. It is more efficient and time saving to place the orders from the stock inventory instead of the Distributors website. By placing the orders from the stock inventory system, they know how many books they have in stock. Another advantage of using stock inventory is that they easily can find out if they have placed an order for a book, instead of having to go into the distributions' website checking there.

3.1.2 Use of Mentor and the website for SentralDistribusjon

Book store chain 1 pays a license to be able to use Mentor. Mentor is used to look up books, if they can not find it in their own system, and to check the status of a placed order to FS, for example to see if there is any delay and what the reason for a delay is. To check the status of an order placed to FS is a bit difficult in Mentor. This is because Mentor displays the orders by an order number they have made themselves, not recognized by book store chain 1.

Since book orders go to the Distribution Central where the book's publisher is connected, as mentioned above, they can not use Mentor to check the status of an order placed to SD, but have to go to SD's website. SD's website does not contain as much book information as Mentor does, but it is free to use.

The IT manager told me that they are satisfied with the routine around ordering books. After they place an order, they usually receive them the next day if the books are available. The status of a book says if the book is in stock, if it may take a while before you can get the book or if you can not get hold of the book. The interviewee told me that the status information is not always updated, which can be annoying if they are waiting for a book that is not coming.

The example below illustrates how they use their system and Mentor:

A customer asks for a book in a store.
The salesclerk tries to find the book in their system first.

a) If the salesclerk finds the book in their system, he can see if they have any left in the store, or in any other store close by. If there are no books left, the salesclerk can check if the book has been ordered.

If there is not placed an order for a book, the salesclerk can place an order himself. If the book has been ordered, he can look up the order at the distribution's website to see why it has not arrived.

b) If the salesclerk can not find the book in his system, he looks it up in Mentor. Since Mentor contains information about almost all Norwegian books, he will probably find it there. The book information is copied into their stock inventory, so he can place an order of the book.

The example illustrates that the book store chain hardly use any of the many functions Mentor

offers. Mentor is only used if they need to look up a book or check the status of an order.

3.2 The interview with book store chain 2

Book store chain 2 had the highest turnover of the three chains interviewed. This chain has somewhere around 150 and 200 stores located in Norway, which say that this is a rather large book store chain. On the 09.09.2004 I interviewed the marketing director from this book store chain. Below is a summary of what the interviewees thought about the information exchange in the book industry today and how this is done at them.

3.2.1 The information exchange today

Client System (CS) is a stock inventory system used at the stores connected to book store chain 2. CS updates the number of books in stock after every sale, and after every order is entered. It is also possible to read and write book information using CS. The book information from Mentor gets downloaded from time to time, and placed on their server. The stores connected to book store chain 2 can then update their systems with the book information from the server. Some book stores automatically get the updates because they are users of the chains server. The information they usually download is technical information about a book, like ISBN, title and price. If the stores want some additional information, they can get this from Mentor themselves. The additional information will have to be typed in by the stores who want it.

If a book is missing in both CS and on the server, the store can get the information about it from Mentor and put it in CS themselves.

Every day a couple of product managers at book store chain 2 use Mentor to look for books they shall use in campaigns. The marketing manager told me that the product managers used Mentor to find 450 books for the fall campaign. The information about the books that are going to be used in a campaign is downloaded.

From the stores website it is only possible to read and order books connected to a campaign. The remaining books the stores sell, the customer have to come to the store to read about or order. The interviewee told me that this is about to change. Before there was not much online shopping and therefore has not this been changed before.

When the stores want to order books this is done directly from their store, without any involvement from book store chain 2. Foreign books are ordered through merchants, while for Norwegian books they use the ordering system in CS. The order will go to either FS or SD, depending on the books publisher. The general rule is to use CS when ordering books, so the stock inventory is updated. However, sometimes orders are placed from the distributions website, which means that the stores have to update the stock inventory.

The interviewee told me that they are working on centralizing the orders of books, so the orders will be sent from one place. A possible solution could be that all orders have to be sent to a centralized location by midnight and orders sent together from there.

3.2.2 About Mentor

The interviewee told me that the book stores in Norway pay somewhere around 9 millions NOK a year for the use of Mentor. At book store chain 2 Mentor is mainly used to look for books to use in campaigns by the two product managers, and by the stores to look up information about books or to check the status of orders placed earlier, if they can not find this in CS.

The interviewee does not think it is right that the book stores have to pay for the use of Mentor, since Mentor is where books are presented. He compared the situation with how it would be if book stores started to charge their customers for coming to their stores to look at books. It is not

only book stores that have to pay for using Mentor, but all users of Mentor including the publishers. In his opinion only the publishers should pay for maintaining Mentor, and not the book stores.

Another matter the marketing director is not happy with regarding Mentor is the ownership. He thinks the owner should be neutral, and not owned by a Distribution Central that again is owned by two publishing companies. Mentor is, as mentioned earlier, owned by FS, and FS is owned by the publishing company Aschehoug and Gyldendal. The publishing companies do not only own FS, but also book stores. The interviewee told me that the publishing companies in theory could see how much and what type of books a store sells, and what kind of books. He is not fond of this situation.

When it comes to the functions and information about books available in Mentor, he still is not pleased. He thinks Mentor contains way too much information about a single book, which makes the book base less efficient to use. The book base is too advanced, and has too many functions. If you want to do a search, there are many ways to do this. The different kinds of search you can do are:

- Search on all books/products: standard search, advanced search, literary prices
- Search on school books: primary and secondary school, adult education, comprehensive school, technical school, all schools
- Search on foreign book bases: wenca, seeling
- News

Within the different kinds of search, you can search for almost all types of information listed about a book. The interviewee told me that they usually search for either ISBN, title, theme, author or published. He does not think it is necessary with all the information registered about a book available in Mentor today. Especially not all the cross categories listed.

3.2.3 Thoughts around how the information exchange could change

An alternative solution for Mentor has been talked about by several book stores according to the interviewee. The interviewee told me that a new book base would make it easier and cheaper for both book stores and publishers. Take the book description for example, where there is possible to do several improvements. Today the publisher writes a book description, which they send to FS. FS then changes the text to make it as neutral as possible in Mentor. However, that's not all. The book description is changed at the book stores as well, to make it more interesting for a potential buyer. In a new book base the publisher could type in the book description directly.

When it comes to the cooperation between publishers and book stores, he feels that this is very important. A book store sits on information important for the publisher and visa versa. For example statistics about sales figures is very interesting for the publisher to know. By informing the publisher of how many copies of a book they have sold, the publisher can plan the books future. Today a publisher is informed about the sales figures for 10 books by book store chain 2. The book store on the other hand is interested in how many books a single book store has bought from a publisher in one year. The format they receive this in today is in a strange format. FS has started to send this in a format which they understand, and SD is working on this as well. He said that this information is important concerning return of books.

3.3 The interview with book store chain 3

The third book store chain I interviewed is a chain with somewhere around 100 stores located in Norway, mostly in shopping malls. This book store chain does not sell only books like other book stores do, but also office supply, other products like cards and gifts. The chain is known as a very profitable chain.

From this book store chain I interviewed a person from the IT-department. This was done on the 08.03.05. The interviewee told me about the systems they use today, and how they get information about books from others in the book industry.

3.3.1 Systems used

All the 100 stores from book store chain 3 uses a system called Megadisc, where information about all books is stored. Megadisc is used to get information about books, and to order books. When an order is placed in Megadisc, the order goes to FS or SD, depending on where the book's publisher is connected.

Mentor is mainly used to look up information like status and description of a book, and to see if the book is in stock. Information about return agreements in Mentor is also used. None of the functions Mentor provides are used by the stores, except searching for books.

Megadisc is going to be replaced with another system at the end of this year or the beginning of next year. They are also planning to do some changes on how the book stores get information about a book. Today each book store has a local database where information about books is stored.

Each day the main office at book store chain 3 receives a text file from FS with information about all books. The file is then checked for updates. If there is any new information about a book or information about a new book, the information is added to their system. The interviewee did not know if they add any additional information than the information from the text file, but he did not think so. Every evening the database from each store receive updates on book information from the main office. The stores give the main office an update on the sales figures each day.

When book store chain 3 is going to change from Megadisc to another system, they will remove the databases placed at every store. They will instead have a main database, which all the stores shall connect to.

3.3.2 The information exchange today

The interviewee told me what he thought about the information exchange with others in the book industry today. When it comes to how the book industry functions in general, he thinks that Mentor is too expensive. He has not any complaints on how Mentor works; only that he does not feel it is right to pay that much for the use of it. They think they pay too much, since they only use it to look up information. Today there is no information in Mentor about how many books there are in stock, which would have been good to know if they for example are going to start a web shop.

Another thing he mentioned was that they do not like that FS is owned by Aschehoug and Gyldendal, since these publishing companies also own several book stores. They feel that they pay FS a lot of money for the use of Mentor and to receive book information. However, since FS is owned by Aschehoug and Gyldendal, it feels like they are giving them money to fund the purchase of competing book stores. He also told me that they are not comfortable with that Aschehoug and Gyldendal in theory can see the orders they place at FS, and how many books they sell.

The text file book store chain 3 receive with book information is a very simple text file, separated

by comma. This means that any changes in the text file, like adding an information type, would require updating their system in order to read the text file correctly. The program they have today reads the text files based upon comma. A standard for sending information would solve this problem, since each information type would be able to be identified, instead of doing the reading based upon the comma separator. A standard for sending the book information is preferable.

Information book store chain 3 miss in the text file today are book description, status regarding number of books in stock, and picture or link to where they can find the picture of the book. If book store chain 3 decides to start a web shop, this is information important to them. He checked how a number of web shops in Norway handle the lack of stock information. None of the web shops he ordered books from gave him any information on whether the book was in stock or not. If the book was not in stock, he would receive this information by mail after a couple of days, and receive the book when it became available.

3.4 Summary of the interviews

I will here give a description over what the book stores do today and what their needs are, based upon what the 3 book store chains interviewed told me:

The book stores are naturally interest in selling books. This is done by having customer coming into their store, or visiting their web shop. For book stores it is important to have information about a book, so they can provide the customers with this information. The book information about books published in Norway does the three book store chains receive from FS and Mentor.

The book store chains think that the information they receive from FS are insufficient and are missing information like picture and book description, and the information located in Mentor are not always updated. It is important for the book stores to have correct information regarding for example the status of the book, whether the book is in stock, being printed or out of production. This information is valuable for the book stores because it is not good business to order a book for a customer, tell her when the book arrives based upon the status information about the book, and then have to tell the customer that the book has not come when the user comes back.

The book store does not need all the book information or functions available in Mentor today. They usually only look up information in Mentor like: book description, status message on the book and if it is in stock, look at the status on order placed earlier, view return agreements and look for books that shall be used in campaigns. Mentor is for some book stores used to look up information about books they can not find in their own system.

Mentor is not used by any of the three book stores to order books, but according to information given by the interview with one of Distribution Centrals are there some book stores those order books from Mentor.

The book store chains interviewed told me that they pay much money to be able to use Mentor. Some of them think that the information should be free for the book stores, and compares this with that they do not charge their customers to come into their store and look at books.

Another aspect of the book industry the book store chains are not satisfied with is the ownership of Mentor. Since Mentor is owned by FS, FS is owned by Aschehoug and Gyldendal, and that these two publishing companies own book stores react two of the book store chains too. They feel that Mentor should have a neutral owner. They feel that they today pay money so the publishing companies can buy more book stores. Another thing regarding this aspect they do not like is that they feel that the publishing companies in theory can look at what kind of books they sell, and how much they sell.

Chapter 4
How the National Library and a publisher exchange information

A publishing company and the National Library (NL) will be interviewed in this chapter. I want to explore who the publisher exchange information with today, and how this is done. The reason for interviewing NL is that there is no information exchanged between the publisher and NL today, which means that NL have to find this information themselves, and adding it to their systems. There has been some discussion earlier whether the book industry should begin to send NL book information, but nothing has been done. I will therefore interview both parties about how things are done today, and how it can be done in the future.

4.1 The interview with the National Library (NL)

The National Library are the organization in Norway where collection, storage and access to Norwegian media are primarily located (NL). There are two locations of NL in Norway today, one in Oslo and one in Mo i Rana.

I interviewed an employee from the Bibliographic Division at the NL on the 17.02.2005. She was familiar with the ONIX standard, but wanted us to have a meeting to look at how the standard could be used for receiving information from publishers.

4.1.1 The questions

For the interview I prepared the following questions:

1. Which companies in the book industry do you exchange information with today, and how is this done?
2. What do you see as the advantages and disadvantages with how things are done today?
3. How would an implementation of ONIX make a difference?
4. What is your information need from ONIX?

4.1.2 Information exchange today at the NL

NL receives today all published materiel in Norway. This is everything from books to magazines, newspapers, electronic documents, music and combined documents. The information about the material is registered in BIBSYS[3]. When a publisher wants to publish a book, they would have to contact NL to receive an ISBN. After the book is published, the publisher has to send 7 copies to NL in Mo i Rana. NL does not receive any information about the book, which means that NL has to find the information in the books, and register it in BIBSYS themselves.

The information registered in Mo i Rana is the most important information like ISBN, title, author and publisher. After this first time registration, more detailed information about the book is registered either in Oslo or Mo i Rana. The important information stored in BIBSYS is transported to Norbok[4], where additional information is added. The 7 copies of the book are stored as followed: 1 copy is stored in a safe place at Mo i Rana, the Depot library in Mo i Rana lend out 1 copy, 1 copy will be placed in the study hall in Oslo, and the remaining 4 are sent to the university libraries in Tromsø, Trondheim, Bergen and Oslo. The registration of information after the first time registration is a more difficult process. Within this registration classifications are set

[3] BIBSYS is a catalogue of everything that exist in Norwegian universities- and research library, and The National Library

[4] Norbok is the National biography; a biography over everything given out in Norway.

and additional information added. The classification of books stored in DNB used to be done by NL, but not anymore. The information about books is stored two places at NL. In BIBSYS the most important information is stored, while in Norbok is stored more detailed information.

4.1.3 How NL wants the information exchange to be

The lack of technical solutions is one of the reasons for not getting information about books from the publishers. Another reason is that their systems and information needs are different. There have also been some assumptions that NL would need much information to cover their need, so there would not be any point sending the book information electronically.

The interviewee told me that ONIX will make it possible for publishers to send information about books to NL. The information NL receive will perhaps not be enough, but they will at least have some sort of control when books get published. Today NL is looking at what information they would like to receive from publishers, and how the information using an ONIX message could be converted into MARC, which is the format used in their systems.

The international ISBN standard is as mentioned in chapter 1 about to change from a 10 digit number to a 13 digit number. Because of this ISO 2108 (International Standard Book Number) has a recommendation that publishers should send the book information to their local ISBN agencies, and that the information should be sent using a format compatible with ONIX. (EDItEUR8)

Information the International ISBN-office (EDItEUR9 2004) recommend that the publishers should start to send are:

ISBN	Product form	Title
Series	Contributor	Edition
Language(s) of text	Imprint	Publisher
Country of publication	ISBN of parent publication	Publication date

Figure 4.1: Information recommended by the International ISBN-office

The information is supported by the ONIX standard. You could say that there is a connection between the ISBN and ONIX standard, since some of the people who maintain the ISBN standard are the same as those maintaining the ONIX standard.

I was told that since the ISBN is changing from a 10 digit number to a 13 digit number, they will have to do some changes with their system. This is a good opportunity to take care of other technical problems as well, so they can start to receive information about books from the publishers.

She did not know exactly which information they would like to receive from the publishers, but she gave me a list of the most important ones:

Title	Original title	ISBN
Author	Associate author	Place published
Publisher	Published (year)	Series

Figure 4.2: The National Library's information need

The ISBN and title are the most important information to receive, along with the book itself. Starting to receive book information it would make it easier for NL to follow up with the publishers if they have not received copies of a book.

A possible advantage of getting the book information electronically is to make the first time registration in Mo i Rana more efficient, or have it added automatically.

It has not yet been discussed how information should be sent from the publishers. A solution she mentioned was that the book information the publishers send to their Distribution Central, could also be sent to them.

Another solution mentioned was not to give the publisher their ISBN, before they have sent NL the book information.

4.2 The interview with a publishing company

The publishing company interviewed is one of the largest publishing companies in Norway. This publishing company publishes around 1000 books a year, including new titles. On the 08.04.05 I interviewed the marketing director from this publishing company. The logistic manager from his department was also present during this interview.

4.2.1 The questions

The questions I wanted to get answered from the publisher were:

1. Which companies in the book industry do you exchange information with, and how is this done?
2. What kind of information about a book is stored in your systems? Is the information stored in Mentor different from this information?
3. What do you use Mentor for?
4. Why is not any information sent to NL today? How could this change?
5. Do you have any thoughts about the information exchange in the book industry today? Is there any room for improvements?
6. How would an implementation of a standard format make any difference for you?

4.2.2 The information exchange today at the publishing company

The marketing director told me that they use a system called Sparta to store information about books. Their Distribution Central collects the book information from this system, which means that they do not have to type in the information on the distribution's website.

The book information for this publisher's books is also stored in Mentor, but Mentor contains some additional information not stored in Sparta. The additional information stored in Mentor is the following:

- FS design a neutral book description, which is only listed in Mentor
- The picture of the book is not sent to FS, but they have to scan it themselves
- In Mentor there is something called book group and product group, where the book group is used by the publishers, and the product group is used by the book stores. The publisher only sends FS information about the book group, while FS have to add the product group.
- Subject, place, time and any other classifications about a book is done by FS, and is only visible in Mentor

Mentor is used by the publishing company to look up information about all books published, and not only their own books. This is useful if they for example want to explore if there are published any books within the same theme as a book they plan to publish. Another thing they use Mentor for is to check that the book published by them has been stored within the right groups. One of the reasons for using Mentor is because this is the only complete base where information about all books is stored.

However, it costs a lot of money for the book stores to use Mentor. The marketing director does not feel it is right that the book stores have to pay to have access to book information. He thinks this should at least be non profit. Another thing the book stores have to pay for is if they want to start a web store. To be able to do this, FS charge a large amount of money.

If the publishing company wants to change some information about a book like the status of the book or price, this is done in Sparta. The updates done in Sparta are being sent to FS, through their Distribution Central, and will then be updated in Mentor. The marketing director told me that they had noticed that FS had changed some of the book information records sent to them, when adding them to Mentor. He do not like that FS do not notify them when they change information.

Neither the marketing director nor the logistic manager knew if NL receives book information or not. Besides their Distribution Central, they send book information to several others. After I told them that NL does not receive any book information from them or any other publishing companies, the marketing director asked if they wanted to receive book information. It did not seem to be any problems if NL wants to receive book information.

The companies besides their Distribution Central that receive book information:

- Bookshops on the web receive electronic book information, for commercial advertisements regarding sales, and for looking up the information from a register.
- The Arts Council Norway also receives book electronic information.
- Book stores and the Norwegian Library Bureau receive a magazine with portraits of authors, cover and information about new books.
- Book stores also receive a chain catalog each year with information about the publishers books
- Some book stores receive a newsletter where information regarding writers or books is presented.
- Book information for commercial use, author portraits and information for the press is also sent to different parts of the book industry.

Since there is no standard format when sending information from them to other parts of the book industry, this is sent in various formats. The reason for this is that the receivers request to receive the information in a certain format. I was told that this is a time consuming job, since they would have to extract book information from Sparta, edit it, and then send it to the company that wants it.

The two persons interviewed felt that the information exchange solution today is ineffective, since both FS and they register information about books into their systems. There is for example no point in having FS scanning the picture of the book, when the publishing company have a picture of the book, and could send this.

4.2.3 How the publisher want the information exchange to be

The two interviewees told me that the information exchange solution today could be more effective than it is today. By for example implementing a common standard in the book industry it would be easier to send information from the publishers to the companies who want it. The publishers then would not have to send the information in various formats, since all companies would use the same format when exchanging information.

It did not seem to be any problem for the publisher to start sending book information to NL, if they wanted to receive book information from them. However, this would become easier if there were a standard, since it takes much time to extract, edit and send book information from Sparta for all the receivers who want it.

Another part that is improved was how the information was sent to companies from Sparta. They think it would be more effective if they found the catalog text they wanted to send, and then sent it from Sparta directly to the receivers who wanted it.

4.3 Summary of the interviews

From the interview with the publisher and the National Library (NL) several problem areas concerning the information exchange today were uncovered.

The publisher told me that there is no standard format used when sending the book information from them, which makes this a time consuming task. The information has to be taken out from their system, edited and structured before they send it to a company. How the information was edited and structured depended on who the receiver was.

An ideal situation for them would be to get the information exported to the companies that requests it directly from their system, without have to edit or structure the information.

Another thing the publisher mentioned was that it is very expensive for the book stores to get this information from FS, which he doesn't think is fear. The book stores have to pay a great deal of money to FS if they want to start a web store, receive information about books or access Mentor.

NL told me that they do not receive any book information today, which they want to change. One of the reasons for why they want to receive book information is that it would make the first time registration done in Mo i Rana more efficient. Another important factor is that NL will have more control over when books are being published, since they today only receive copies of the book from the publisher when it has been published. Since there is no control of if the publisher has sent in copies of the book, NL will not get to register the information about the book if the publisher forgets to send it.

NL had heard about a standard called ONIX, which is used to exchange book information using a standard format. The ONIX standard can be converted to MARC, which is the format used in their systems. NL wants to start receiving book information from the publishers on a standard format, to make it easier to add it directly to their systems.

Chapter 5
Requirements and needs

This chapter will begin by listing the problems discovered during the interviews, before dividing the problems into 3 problem areas. Each of the problem areas will be discussed, but only one of them will be focused on in this thesis. The requirements and needs concerning this problem area will be described in the end of this chapter.

5.1 Problems with the information exchange solution

Here a list of the problems discovered during the interviews with the three book store chains, 2 Distribution Centrals, the publisher and the National Library (NL) will be listed:

Information and functions in Mentor: The majority of the companies interviewed thought that Mentor contains too much information about a book and too many functions. They did not use all the functions and information, but still have to pay for it. One of the interviewees wanted a more efficient book base, where there is less information and not so many functions.

Updated information: One of the book store chains told me that Mentor does not always contain updated information, like which status a book has (in stock or not).

Additional information: One of the book store chains wanted information about the availability in stock for a book. He also wanted to receive book description and link to a picture of the book when he receives the book information text file from FS.

Inefficient information exchange concerning the Distribution Centrals: One of the Distribution Centrals told me that it would be easier to send book information to book stores using a standard format. Both Distribution Centrals meant a standard would make it easier to exchange information between SD and FS.

Time consuming information exchange from the Publisher: The publisher felt that the information exchange solution today is time consuming, since they have to send book information to different places using different formats. This could change by using a standard.

Inefficient registration of information: The registration of information is done in too many places today, which makes this process time consuming and inefficient. The publisher thinks FS should only add classifications and groups concerning the book stores, which is information they do not provide.

Information Registration at FS: Some of companies told me that they do not think it is necessary to have FS classify, edit and add all the information about book, which they do today. There is no point in having FS changing and adding information and about a book, when the information the publishers send can be used.

Receive information at NL: The National Library does not receive any book information electronically from the publishers today. They want to start receiving this information, and want this sent using a standard format.

Ownership: Some of the companies interviewed did not like that the Norwegian book database (DNB) and the book information stored within it is owned by a Distribution Central, which again is owned by two publishing companies.

I was told by a couple of the companies interviewed that the publishing companies in theory could see what the book stores order, and how much they sell. This is not a good situation since the publishing companies own two book stores themselves. The interviewees who this problem thought that the Norwegian book database should be controlled and maintained by a neutral part of the book industry, and not by someone with an interest in how things are done.

Costs: The three book store chains I interviewed told me that they have to pay much money to use Mentor. One of the interviewees told me that he does not thinks it should cost anything for them to get information about books. The publisher interviewed also thinks the book stores have to pay much to get hold of book information.

5.2 Discussion

In this section the problems listed in the previous section will be discussed.

I divide the problems into three main problem areas based upon what type of problem it is:

- Problem area 1: Standard
- Problem area 2: Information and function need
- Problem area 3: The Norwegian book database (DNB)

This thesis will only focus on problem area 1 in the following chapters, but I will still have a discussion of all three problem areas below.

5.2.1 Problem area 1: Standard

Because there is no standard format used in the book industry in Norway today for exchanging information, the process becomes difficult and time consuming for all parts. The publishers have to make sure that the information is stored in a specific way so the receiver will understand what the publisher is referring to when they send the book information.

Most of the publishers connected to SD register the book information through SD web. By registering the information on one of SD's forms, the publishers do not have to think about how the information is structured. However, it is time consuming for the large publishers to have to type in the information of all new books to SD web, since they already have this information stored in their own system. It would for them be easier to send over the book information electronically, directly from their system. However, if the publishers start to send information to SD electronically without using any standard, SD would receive book information of different structures, using various types of format. Adding this information to their systems would be difficult. A standard is therefore essential for them if the information should come electronically.

The publishers also send book information to other parts of the book industry, and according to the publisher interviewed, no standard format is used. The companies that receive the information want the information in a specific format, so the publisher has to structure the information as requested before they send it. This is a time consuming job, and would become more effective if all parties used the same standard format when exchanging information.

SD sends the book information received from their publishers to FS, who put the book information in the Norwegian book database (DNB). FS does not add the information directly to DNB, but type it in after they have checked it, and added classifications and other missing information to the book. FS wants to receive book information from SD by a standard format, and to send the book information to the book stores using a standard format. This would be much more efficient since the information from SD would be added directly to their system, and the information to the book stores would be sent using the same format. SD feels that implementing a standard would make it possible for them to add book information directly into DNB, so FS will

not have to use time and recourses to do this.

The National Library (NL) also has to type in the book information to their systems. This is because they do not receive any book information from the publishers, only copies of the book. The interviewee from NL told me that they want to start to receive book information electronically. To make this possible, the publishers would have to use a standard format, so the information is easily added to their system.

The book information is therefore typed in at least three times; at the publisher, at FS and at NL. Some publishers even type in the information twice, since they have typed in the book information to their system and again when sending it to SD using SD web.

SD thinks that an ideal situation would be if the publishers were the only one part registering the book information. They do not think it is necessary to have the information typed in more than one time, and since the publishers are the part that first register and send the information, they should be the one that type in the information. The other parties should be able to receive and forward the book information.

The interviewee from FS told me that they have to add, classify and quality check much of the book information sent over by the publishers, since the information sent over are sometimes wrong or missing. The information they receive are then added to their own system after they have done modifications with it. She said that she thinks FS are better suited than the publishers to classify books and add a neutral description about the book. Especially since the staff there are librarians, and therefore have knowledge on how to classify books.

Further she said that if the publishers start to send more detailed book information than they do today, it is not certain the FS' job would become easier. The knowledge among the publishers are various, which would mean that FS would have to do this anyway when they quality check the information. There is no point in having the publishers send detailed book information, if FS are going to check it again when they receive it.

Having FS add, classify and quality check all the information sent over from the publishers is a time consuming job, and not really necessary. The publishers spend much time in registering information about their books, and if is not an efficient solution to have FS go through all the information again and check if it is correct. Even if there are publishers who do not the knowledge regarding classifying the book, they should be able to send over the correct facts about the book.

Having the information added in DNB directly is neither an appropriate solution, since the information FS receive does not cover all the information that are going to be added in DNB. Information like classifications, neutral book description by FS, picture, product group codes for the book stores are examples of information FS add after they receive the information from the publishers. The publishers do not have the knowledge to set this information. If the information is added directly into DNB would this information most likely be wrong or missing.

Choosing a combination of the two solutions are the best. The publishers could send FS the most important book information, which are facts about the book. FS are not going to quality check this information, but only add classifications, product groups for the book stores before adding it to DNB and other information agreed upon that the publishers are not going to send. The publisher I talked to told me that they do not send over classifications, picture, or product group for the book stores today, but facts about the book itself. By doing it this way, FS would be able to add information difficult for the publishers to know. Another advantage would be that the time spent on each record would go down since the same type of data is not entered twice. The publisher I talked to told me that they have the picture of the book stored in his system, and can therefore send this over to FS so FS don't have to scan it. The neutral description of the book is not

necessary to have stored, since the publishers have already designed a book description.

However, there is no point in changing who adds different kind of information if there is no standard format used for exchanging information. If FS are going to stop typing in all the information, they must receive the information in a common structure. The same goes for NL. If they start receiving book information and adding it directly to their system, the information must be sent using a standard format. According to a document NL gave me (Nasjonalbiblioteket 2003), they want to start receiving book information from 2007, when the new ISBN comes. They want the standard to be ONIX or another standard compatible with MARC, which is used in NL's systems.

Based upon the response from the companies I interviewed, a standard format for the exchange of information is preferred. The Distribution Centrals and the National Library (NL) had already heard about a standard called ONIX before I talked to them, and were looking at how the standard could fit together with their systems. I talked to the publisher and one of the book store chains about implementing a standard in the book industry as well, and they agreed that implementing a standard would make the information exchange more efficient and better then it is today.

A standard could be implemented in several places, and implemented by many organizations. I will give a list of both below. If the book industry decides to standardize, they have to look at where the standard are going to be implemented first, and by whom. If a standard are going to be implemented all places at once, or one place at the time, have be discussed and agreed by the book industry.

Places to implement a standard based upon what the interviewees told me:
- From publisher to Distribution Central
- From Distribution Central/publisher to the National Library
- From publisher to companies within the book industry who wants book information
- From SD to FS
- From FS to book store

The companies that have to implement the standard:
- Publishers
- Distribution Centrals
- Book stores
- The National Library
- Other parts of the book industry who the publishers send book information to

This thesis has focused on how to improve the information from publisher to book store through the Distribution Centrals, and not the information exchange between other parties in the book industry like libraries and printing agencies. However, if the book industry implements a standard format, this would also affect this part of the industry to.

Since the book information is registered the first time by the publisher, and then sent it to their Distribution Central, the publisher and the Distribution Central will be the first to adopt the standard if not all the companies are going to implement the standard together.

There are many small publishing companies, who publish only a couple of books. These publishing companies might have problems with implementing a new standard format for sending book information to their Distribution Central, since they may not have the money to implement it. For these publishing companies it might be best to let them continue to register the book information into SD web or Mentor, depending on which Distribution Central they are using. Figure 3.1 that the marketing director at SD gave illustrates this. From the figure you can see that the publisher can choose between 3 ways of sending their book information to SD, where the two

main ways would be using SD web registration or sending the information electronically using a standard format. As long as the Distribution Central receives the information either way, the information will be added to their system.

The book store chains I interviewed were more concerned with the information and functions available in Mentor and how they could be improved, than how they receive the book information from FS today. This is because the only information they exchange with other parties in the book industry is when they receive book information from FS to put in their systems. They are hardly affected by the lack of standard like the publishers and the Distribution Centrals are.

One of the book store chains told me that they receive a text file with book information. There is not any industry standard for the text file, and the book store chain read the file separating the information by comma. A standard would make it easier to read and understand the book information sent over. Changes in the text file would not be any problem either. If an extra field is added in the text file today, the book store chain would have to update their system to read the text file correctly. By using a XML standard their system could read the file sent over based upon identifiers to each information type, and not by comma. Changes done to the set of information would not be any problem either, since the information would be read and stored based upon the identifier to the information.

Based upon what the Distribution Centrals told me in the interviews, are they both interested in a new standard format for exchanging information. So besides staring to implement a standard between the publisher and Distribution Central, the implementation has to be done between the two Distribution Centrals. An ideal situation would be to implement the standard these two places at the same time, since the Distribution Centrals are already doing changes with their systems. The last place to implement the standard of the three company types would be at the book store, since they are not that affected by the lack of standard, and the publisher and Distribution Central have to make sure that the everything is working between them before the book stores will start to receive the information using a standard format.

When the information exchange between the publisher, Distribution Central and book store is standardized, it is time for the remaining companies that the publisher sends book information to standardize. This includes NL, since they want to start receiving information about books from the publishers.

Adopting a standard would not make any difference regarding which information or functions Mentor provide. However, adopting a standard would make the information exchange more efficient, since FS would not have to spend that much time adding the information into DNB and quality checking the information. A reduction of the price for receiving information from FS and for using Mentor would therefore not be unlikely.

In section 5.5 I will make a list of the requirements the companies interviewed have regarding a new standard.

5.2.2 Problem area 2: Information and function needs in Mentor

The problems listed regarding information and functions in Mentor and information that one of the book store chain were missing, in the file they receive from FS, will be discussed here.

Mentor contains a great deal of book information and many functions, used by book stores, libraries and publishers. I do not know exactly which functions the companies use in Mentor. I have been told that some company’s order books using Mentor look up information about orders placed earlier and look at the return agreements. The book store chains told me that they mainly use Mentor to look up information like description and status about a book, if they can not find the information in their own systems, or to look for books that they are going to use in campaigns.

The publisher told me that they use Mentor to look for competitive books on the market within a subject, to see if they are going to publish a book or not.

From the interviews I did with 3 book store chains, I got the impression that it is very expensive to use Mentor. The book store chains are not happy with paying for viewing information about books in Mentor, especially since they do not use many of the functions, the information is not updated and they do not need all the information stored. By looking at the job FS do with the book information; classifying it in several groups and quality checking all the information, this is a job that takes very long. It is probably the reason it cost so much for the book stores to use Mentor.

The interviewee at FS told me that it costs to produce some of the additional information about a book like description and links between books they add to the information they receive. The present of this information depends, according to FS, on if the users want it or not. I do not know when FS had their latest marketing research to see what the information need among the users is today. Based upon what the three interviews I did say, Mentor does not have to contain all the information and function it does today. It would be smart to do a marketing research among the users to see what type of information and functions the book stores and other users need from Mentor. The information most users find unimportant can be removed, so the costs could go down. Later in this chapter there will be listed the information two book store chains find important. This information might not be representative of all book stores and others needs regarding Mentor, but gives a picture of the most important information necessary to have about a book.

From the interview with FS I was told that they had received positive feedback from book stores regarding the functions and information available in Mentor. The three book store chains I interviewed told me that they did not need all the information or functions Mentor provided. Even if the three book store chain I interviewed presents a third of the total turnover among the book stores, I do not feel that their answers can represent what all the book stores need. By doing a marketing research would make it easier to see what kind of information and functions the book stores use and need in Mentor. The information the book stores need to receive from FS will then also be uncovered.

There is no point in providing information and functions in Mentor if the book stores do not use them. Likewise the information and functions should not be reduced in Mentor before it is known what the book stores really use and need. There is no point in classifying a book several ways or adding other types of information if there is hardly anyone that uses it, especially since it costs a lot of money to add. Books are today classified and categorized based on title, author, subject, product owner, publisher, series or not, book review, picture, index, price and availability. (Forlagsentralen1)

5.2.3 Problem area 3: The Norwegian book database (DNB)

The last problem area uncovered from the interviews was concerning DNB and how the book industry functions in general. From the interviews I got the impression that several companies were not happy with the owner situation of DNB, and the high price of using Mentor.

If a new standard are implemented in the book industry there will have to be changes in the way they exchange information with each other so their systems will fit with the new standard. This might be a good opportunity for doing some additional changes with how the book industry functions as well. Since the information is registered several times by different companies I feel that there is room for a more efficient solution. In this thesis I have only looked at how a XML standard could be implemented in the book industry, and not explored other changes that could be done in the book industry.

Even if I am not going to explore how other changes for the book industry, I will below suggest how the information exchange in the book industry could function. The solution will among other things contain changes in the way DNB is maintained and how book stores look up information. The solution is based upon information given in the interview with NL, and as an answer to the problems the companies interviewed had.

It is important that companies in the book industry discuss the options available regarding the owner situation of DNB, since some of the companies interviewed focused on this as a solution not preferable. They think the owner of DNB and Mentor should be neutral. FS will most likely want to continue to maintain DNB and Mentor, even if the others feel that the owner of these databases should be neutral. It is however important that the book industry at least discuss what could be done, and see if they come to an agreement. It is better that the book industry discusses their options, and try to find a suitable solution they all agree upon, than getting a competing database for storing books developed. A competing database would lead to not having one place where information about all books is stored, which is one of the things the book store chains liked about Mentor; that it contain information about all books published in Norway. If the book industry agrees upon getting a neutral organization to maintain the book information of all Norwegian books, a good candidate is NL. NL has good knowledge about the information stored about books, since they do it themselves today. NL also knows how to classify a book, since they used to do this for FS. I have designed how it would be if NL started to be in charge of a database for storing book information of Norwegian books.

I was given a document (Nasjonalbiblioteket 2003) from NL, where NL discuss if they should develop a metadatabase with book information. NL could get all the publishers to add their book information to this database, by not giving the ISBN to them before they had received the book information. Since the publishers have to get the ISBN in order to publish a book, they would have to register the information, which means that NL would have a complete database over books published in Norway.

If FS would lower their fee for the companies who use Mentor and receive book information, it would be a step in the right direction. The companies interviewed do not think Mentor is a bad place to collect information, but they feel it contains more information than they need. That leaves the ownership situation, and how the book industry and FS could improve the owner situation.

Since several of the companies interviewed told me that they do not like that the owner of DNB and Mentor not is neutral, an alternative is that NL could have control of how the information exchange would be, and the information sent from the publisher to the book store.

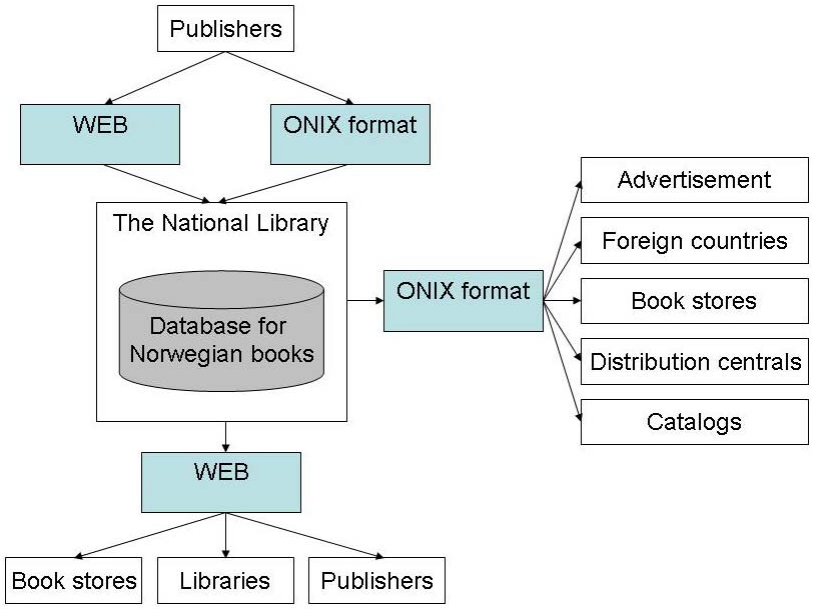

Figure 5.1: How the book industry could be connected

From the illustration above the National Library (NL) are set to have control of a database for Norwegian books. NL is a neutral part in the book industry, with the knowledge about books, important information about books, and how to classify books. I was told in the interview with NL that they used to be in charge of classifying books for FS. Today FS does the classifying themselves. I have not investigated if NL would be interested in controlling and maintaining a national database. I have chosen them in this solution to illustrate how a neutral part of the book industry is a potential candidate to maintain the information about Norwegian books. An alternative to NL are to create a new organization, where the organization only are going to maintain the information about books published in Norway. An organization like this would probably need 2-3 persons working there, according to an IT manager from the book industry, in case someone are sick, on Holliday or something else.

It is important to look at how much it would cost for a new organization to control and maintain a national book database, compared to having an existing organization like NL do it, or having FS continue doing it. It is important that the cost of maintaining a national database not are very high, since the high cost using Mentor and having FS spending time classify and quality check the information is one of the things the book store chains were not happy with. Giving a new organization the responsibility of a national database would perhaps cost more then having an already existing organization like NL do it, since the new organization have to have at least 2-3 people working there to function, while NL would not demand that many people to work with this. If the information is going to be added to the national database more or less directly, their job would be more to make sure that everything is working. If the book industry decides to have the information classified before they are added to the national database, NL already know how this is done, while the staff at a new organization might not have the knowledge about this.

When it comes to the information sent from the publisher to the national database, this could be done by either sending the information to their Distribution Central like they do today and the Distribution Central would forward the information after they have stored the information they need in their system. The second alternative is to send the information directly to the NL and DNB, and they will add it to the national database.

The figure illustrates that the publishers register book information in two ways; either by typing in the information on a website like they do today, or by sending a file with the information.

Whether there is a need for someone to add extra information or classifications, or quality check the information sent over is up to the book industry to decide. By doing a marketing research among book stores as mentioned earlier, the information and functions that the book stores need and want will become clear. After this has been set, the publishers have to agree upon if they are going to deliver all of this information themselves, or if NL or another organization should add the additional information. Classifications for example could be added by NL, since they have better knowledge of this information than most publishers.

After the information has been added into the database, several parties in the book industry will be able to get out book information in a structured standard format. NL and the book stores for instance are interested in getting the information about books electronically to store in their own systems. By using a standard like ONIX, the information could easily be added to their systems.

The figure illustrates that the information from the national database goes to companies in foreign countries and marketing material for catalogs, besides book stores, libraries and Distribution Centrals.

Whether the Mentor database and view towards library, book store and publisher will be kept, depends on the need among the users is and what the book industry decides. If they do not need all the functions and information available and many changes in Mentor have to be done to satisfy the users, it is perhaps better to design a new web view towards the users, based upon the information in national book database NL maintains. If the marketing research informs that the users are satisfied with Mentor and the information and functions within it, the Distribution Central can receive the book information from the national database, which can be added in Mentor. In the figure above I have illustrated how a web view towards the book stores, libraries and publishers would be connected directly to the national database.

5.3 Requirements for a new standard

If the book industry implement a new standard, it is important to make sure that the standard will support the companies' requirements.

5.3.1 The book store chains requirements

If a standard is implemented and the book stores should begin to receive book information sent using a standard, it is important that it will support the information need they have. Two book store chains I talked to gave me the following list of information they need to receive about a book, and place in their system.

Title	Original title
Subtitle	ISBN
EAN	Subject
Language	Published
Edition	Amount of pages
Price with tax	Price without tax
Sales price time	Minimum order amount
Tax obligated	Picture
Book description	Audience level
Format of the book	Weight of the book
Binding	Author
Publisher	Distributor/Supplier
Telephone distributor	Fax distributor
Email distributor	Zip code/Place distributor
Product owner	Editor
Illustrator	Designer
Translator	Dewey
Series number	Series
Product group	Book group
Literature form	Literature type
Status message	Statistic group

Figure 5.2: Two book store chains information need

The information listed above is information two book store chain told me was important to receive and store in their systems. Much of this information are searched for and informed to the customer if they ask for a book. They also mentioned that they would like to receive other types of information as well, such as the place and time a book is related to. However, they feel it is important to separate between information that they need to have, and information nice to have. If I had asked more book stores, there might have been additional information in the list above.

5.3.2 The National Library's requirements

The National Library (NL) does not receive any book information today, but they would like to change this. NL gave me a list over important information they want to receive in figure 4.2. The information listed by the NL is a subset of the information listed by the two book store chains. The information has to be sent using a standard like ONIX, which can be converted to MARC, which they use.

5.3.3 In general

In general, a standard have to be both open and flexible, since companies use different kinds of systems, and a standard have to function well together with all of them. The information need among users could change with time, and the must be easy to evolve in order to meet new requirements in the future.

Chapter 6
Standards

This chapter will describe the importance of standards, and the challenges with standardization. I will also look at how the book industry in Norway could standardize the information exchange solution.

6.1 What is a standard

A standard is something that a group of people agrees upon, recognized by the whole group, and by people outside the group.

There exist several types of standard; private, public, European and International are some of them. A private standard is used only by the organization that developed the standard, while a public standard can be used by several organizations. If a standard is European, it has been developed by the European Committee for Standardization (CEN), where the members are national standard bodies of the European Union. An international standard is produced by the International Organization for Standardization (ISO), where the members are national standard bodies over the world (BSI 2005).

An alternative to a standard can be when companies agree amongst themselves how things shall be done. To agree with others on how they shall communicate is not an appropriate solution if there are many companies involved.

The figures below is based on a figures drawn in class at the IS and Standardization lecture for the inf5250 course (INF5250 2004), and illustrate the difference between using agreements and using a standard.

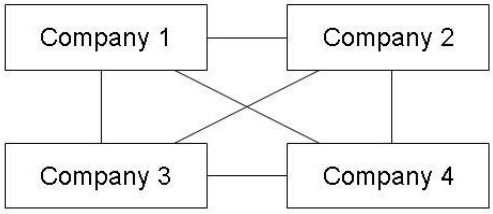

Figure 6.1: Use of agreements

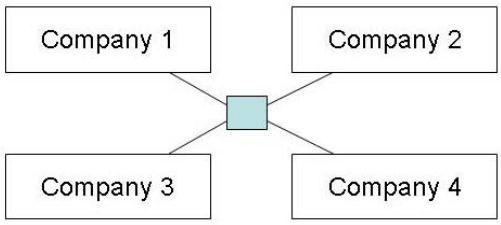

Figure 6.2: Use of standard

Fig.6.1 illustrates how companies have separate agreements with the other companies they communicate with. In fig.6.2 the companies have agreed upon a common standard that all the companies use. Using a standard instead of having agreements with others is much easier, especially if there are changes in the environment. If a new company were to be added to the group, this would be more complicated in an environment where there were only agreements between companies. The new company would then have to agree with all the companies on how to communicate with them. This would not be the case if the companies were using a standard, since the new company would have to start using the standard that all the others are using. Same goes for changes in what is being communicated from one place to another. Instead of doing the changes to the standard like the companies in figure 6.2 could do, the companies affected by the changes in figure 6.1 would all have to update their system, which would be ineffective when changes was needed.

According to Hanseth is a standard is preferable compared to having agreements between companies. The primary argument for using a standard is that designing and maintaining a larger number of agreements between computers and applications is extremely expensive compared to using a shared standard (Hanseth).

Hanseth et al (1996) claims that agreements between companies that communicate is only possible when there are only a few companies. When there are many companies that communicate it is neither cost-effective nor possible to have a large amount of agreements between all of them. A standard is then preferred (Hanseth, Monteiro et al. 1996).

The book industry in Norway exchanges information about books according to figure 6.1 today, where the companies have made agreements on how the document are structured, and which format to be used. This is not an ideal solution since there are many companies within the book industry.

According to the two articles above, a group of companies of a larger size have to use a standard when communicating with each other. The information exchange in the book industry is today time consuming and ineffective because of the lack of a standard. The publishing company I talked to told me that they have to edit their book information in various format, before sending it to different parts of the book industry. One of the Distribution Centrals told me that they receive book information in various format, and therefore have to type in the information they receive to their system, which is a time consuming job. The book industry is of a large size and much information is sent from one place to another. A standard are therefore used to make this process more effective. The publisher would then be able to send off information using a standard, instead of making sure the information are structured so the receiver will understand what the information is about, and the Distribution Central would be able to add the information directly to their systems.

A group of companies that wants to standardize a process, which has been based upon agreements, have the following two options:

1. Find a standard which would meet their requirements
2. Design a new standard that would support their needs

It is important that the standard a group of companies choose, meet their requirements and need, since it costs a lot of money to change from one standard to another at a later stage.

I will in the next section look at what a standardization process is, and how the book industry can standardize their information exchange process.

6.2 A standardizing process

Standardization is a process where companies choose a standard they to use for example to communicate with others. There are several challenges or important factors that companies must reflect on if they want to standardize. Examples of this can be: choosing, developing and the implementation of the standard.

When choosing or developing a new standard, it is important to make sure that the standard is both open and flexible.

After a standard is adopted, further development in the organization or the group is based on the openness of the standard. If a standard is open and can easily be combined with other information systems, further development later on will not be any problem. If the standard does not go very well with other information systems, it will most likely be difficult to do modifications with the

information system later, or choose a new information system.

It is also important that a standard is flexible to be able to overcome challenges that the developers have not foreseen. Challenges could be everything from problems in the initial face not for seen, or in the environment on a later stage. An example of a flexible standard that had to meet challenges in the environment is the Internet standard. The Internet standard had to go through several changes since it first came, because of changes in the environment. Since the number of users on the Internet has grown tremendously since its introduction, the standard has had to support the increased number of users, the IP standard has to be upgraded from IP version 4 to 6 (Hanseth, Monteiro et al. 1996). This is a challenge that the developers did not have for seen when they developed the standard, but because the standard they developed was flexible, it will overcome this type of challenge.

Other challenges when implementing a new standard are that companies within the group do not want to adopt the new standard. They may be satisfied with how the things are done, or do not have the money or the knowledge to make the modifications necessary in order to adopt the new standard. In order to make the implementation of a new standard successful it is important to include all the companies in the adoption process, and assist them with the necessary modifications. There is no point in having only a couple of companies adopting a standard, since the advantages from using a standard would first come when all or at least most companies us it. Ways to help small companies who do not have the knowledge or resources are to assist them with the implementation.

6.3 Standards available for the book industry

An IT manager from the book industry told me that if the book industry decides to standardize the information exchange is there only one real standard and that is the ONIX standard.

The IT manager told me that the book industry in Sweden adopted a sort of standard for the information exchange process a couple of years ago. The standard is user defined and does not have a name. According to a document given by the IT manager about this standard (SwedenStandard), has it 66 elements describing a book. After exploring the elements used in this standard, I saw that most of them were in the ONIX standard (EDItEUR6 2003). However, there are somewhere around 250 elements in the ONIX standard, describing a book/product, while the Swedish standard has only 66 (SwedenStandard). Keeping in mind that the information need among the users may change with time, a standard that support much more information then the need today is to preferable. Since the standard used in Sweden is user defined and only fur fill the requirements the users have today, it would have to be enlarged if the information needs increasing.

The ONIX standard that I will explore in the next chapter is the only standard that seems to have the potential. It was introduced in 2000, and has since then become an international standard and been adopted in several countries.

In the general requirements in section 5.3.3 I wrote that it is important that a standard is open and flexible. The ONIX standard is flexible since it has gone through several changes since it came; to meet new features companies felt were missing. It has been adjusted to meet new requirements, which therefore makes it flexible. The standard is also open, since it is based upon XML, which is platform independent. This means that everyone can implement the standard, independently of what kind of systems they use.

By adopting an international standard used by several countries instead of a user defined standard designed especially for your country, have several advantages. Since the international standard are being used by companies in several countries, it contains most likely all the features necessary to

have in a standard. Since the standard is being used and therefore been tested, it probably do not contain any errors. An international standard also makes it easier to do business with companies in foreign countries. Two of the book store chains I interviewed told me that they order books from foreign countries. By adopting an international standard would make it easier to import and export books, if the countries you are cooperating with use the same standard as you. For the book stores would the implementation of the ONIX standard make it easier to add the book information from foreign companies into their system ,if the companies they order books from also have implemented this standard. The international standard would make it easier for the publishers to export information about their books to foreign countries.

The format used in Norway today is designed using a bottom-up strategy. The standard is small when it is first implemented, and increases with time after additional information need arises.

The ONIX standard is designed using a top-down approach. The companies choose what kind of information they want stored about a book among all the book information elements the ONIX standard provides. The standard supports the need a company has even if they have small or large information need. By using the ONIX it would be easy to adjust the information you want stored about a book, since the standard contains all sorts of elements with information about the book. If a companies information need increased, this is not a problem, since the information most likely already are supported by the standard.

6.4 A standardization process for the book industry

If the book industry decides to standardize the information exchange process, the following I have designed the following guidelines that the book industry should do:

1. The first thing the book industry has to do is to create a small group with representatives from publishing companies, Distribution Centrals and book stores. The representatives within the group would be responsible for talking to the rest of the book industry, based on which part of the industry they belong to. The group will discuss and find the requirements that are important for them, and try to find the standard that support these requirements. It is important to establish the requirements from the different part of the book industry at an early stage to make sure that the standard they choose can support them. Before choosing to go for a standard available, they would have to investigate the standard they got to choose from, and if they are any good. There is no point in developing a new standard, if there is a standard on the market that can be used and support their requirements. The group has to include all parties that will be affected by the standardizing process as much as possible, to make sure they agree upon the choices being made. Including everybody in the standardizing process will make them more positive to a new standard.

2. If they decide to adopt a standard already available, like the ONIX standard for example, they would have to go through the standard to look more closely how the standard supports their requirements. By implementing the standard they can do some testing, to see how it works and how it supports the requirements important for the book industry. If the standard needs some adjustments in order to support all the requirements, this has to be presented for the organization who manages the standard. EDItEUR is the organization to contact regarding modifications with the ONIX standard.

3. After testing the standard and making it ready for the book industry, a set of guidelines regarding how the standard is implemented have to be designed. It is important that companies that need it will be assisted, since not all companies have the money or knowledge about implementing a standard. A meeting where the standard is presented, and where they tell about the standard’s features is also necessary in order to show the other companies why it is important to implement a standard.

Chapter 7
The ONIX standard

This chapter will be about the international ONIX standard. The standard was implemented in 2000, and has since then become an international standard. How the implementation of the standard is coming along in countries has also been explored in this chapter.

There have been several versions and revisions of the ONIX standard. I have explored release 2.1, rev.01 of the ONIX standard, but also collected information from rev.02. Some of the documents I will refer to in this chapter and in the XML chapter is from the ONIX standard downloaded at EDItEUR (EDItEUR1).

7.1 About ONIX

The FAQ at BISG (BISG1) shows that ONIX, short for ONline Information eXchange, is a standard file format for information exchange and representation in electronic form in the book industry. It was the Association of American Publishers (AAP) who came with the preliminary version of ONIX in January 2000. Only a couple of months later the new version of ONIX was introduced. This version was developed by the international organization EDItEUR in cooperation with Book Industry Communication (BIC) from UK/Europe, and the Book Industry Study Group (BISG) and AAP from USA.

There have been several versions and improvements with the ONIX standard since it was introduced in 2000 (EDItEUR2). Release 2.1 of the standard came in June 2003, and it became an international standard. Release 2.1 was a result of feedback from users in several countries. Release 2.1 has full upwards compatibility from release 2.0. The ONIX for Books Product record format has several additions in the last release. If users do not need these new additions, they do not have to do anything else than change the release number in the header of the ONIX message.

In February 2005 an updated code list of rev. 02 came (Editeurnews 2005). The reason for this update is the increasing number of countries adopting the standard, and it therefore needs some additional codes to fit with their systems.

The responsibility of publishing and maintaining the ONIX standard falls to EDItEUR in cooperation with the companies mentioned above. EDItEUR is a non-profit organization located in London (EDItEUR3). The organization consists of 90 members from 17 countries. EDItEUR's main job is to coordinate the development, promotion and implementation of Electronic Data Interchange (EDI) and other e-commerce standards in the book- and serial sector.

This chapter is about the ONIX standard for books, but there are also other areas of the standard. From Martin's presentations (Martin 2004) it is written that other areas of the ONIX standard are:

- For serials
- For video and DVD
- Other ONIX applications: MEDRA DOI registration metadata and ISBN registration metadata

ONIX was developed to solve 2 problems (BISG1):

1. The need for more information about books online
2. The many format specifications the players in the book industry used

Having a standard format for sending information makes the whole industry more efficient. Publishers do not have to format their book information depending on who the receiver is. Book sellers, more easily, get information which fulfills the customers need. It is a proven fact that the more information a customer has about a book; the more likely it is that the customer is going to buy the book. ONIX has made it possible to send much more about a book. It is therefore up to the publisher how much information they want to give the customers.

The Internet presented a new channel for selling books, and was an important factor to the creation of ONIX. There had been problems getting book information from publishers to book stores, because no standard for sending the information exists. A standard format the publishers have to use when sending their book information was therefore agreed upon.

In 2001 the following objectives were hoped to be achieved with the ONIX standard (CoverPages 2000):

- To cover not only books but also other media which are published and distributed through the book industry
- To meet the practical information needs of all sectors of the industry, including but not limited to online booksellers
- To reflect the realities of national and international rights, distribution, pricing and availability
- To be usable in a multilingual marketplace
- Very importantly, to incorporate the core content which has been specified in national initiatives such as BIC Basic and APP's ONIX
- Finally, to build where possible on what EDItEUR's EPICS and the <indecs> Project has done to establish sound models for metadata in a future electronic environment.

The intention with ONIX was to develop a universal international format, to enable all publishers to exchange information about books. ONIX can be viewed as a language or format the publishers could use when distributing electronic information about their books to wholesale, retail and online booksellers, distributors and other publishers. (BISG1)

Twice a year the ONIX international Policy Committee meets in London and Frankfurt. This Committee consists of 2 members each from ten countries, and is the one who sets the standard.

7.2 How ONIX work

According to EDItEUR companies that want to implement the ONIX standard would have to organize their book data into ONIX-specified fields and store it in a database (EDItEUR4). Changes to their existing database can also be done. In the database chapter different database solutions will be explored to see which one is best fitted to store book data based upon ONIX. After the company has organized their book data, they can use an XML software tool and the ONIX DTD to organize and tag the data.

When a publisher is going to send book information to their Distribution Central, he could use an ONIX message to do this. The ONIX message is built upon information that is in the publisher's database. If sender and receiver are both using the ONIX standard, the receiver will know what each field represent, and translate the data into information used in a web page or in other systems. I will show how an ONIX message is built in the next section.

The ONIX standard defines both a list of data fields about a book and how to send the data in an

ONIX message.

To transmit an ONIX message you can either use the Internet, by way of an email attachment, FTP (file transfer protocol), CD-ROM or any other way of moving data from one computer to another. The receiver of an ONIX message uses the same tool as the sender to verify the data's integrity. The receiver then gets the data translated into what you can see on a webpage.

7.3 An ONIX message

The intro file (Intro) show that ONIX consist of over 250 data elements divided in 40 groups. 25 of these refer to "Product records", 7 to "Main series" and 8 to "Subseries records". There are only a couple of elements mandatory, while most elements are optional. Even if the elements are optional, they have to be included in the ONIX message. Most of the data elements consist of text, but reference to other multimedia files, such as images, audio files is also used.

The ONIX message is a set of data elements defined by "tags" which are written in the computer language eXtensible Markup Language (XML). The message conforms to a set of rules given in the ONIX Document Type Definition (DTD). A more detailed description of XML and DTD will come in the XML chapter.

An ONIX message consists of one Header element that shows who is sending the message, and when it is sent. You can define the Header element in two ways, depending on whether you are going to use a DTD or not. The ONIX message also consists of 1 or more Products, MainSeries or Subseries elements. There will be an illustration of an ONIX Product Information Message in the XML chapter.

7.4 Advantages with implementing ONIX

There are several advantages by implementing the ONIX standard. I will here give a description of the ones I find most important based upon articles and presentations I have read.

The first version of the ONIX standard arrived in 2000, and has been used and tested by companies since then. The version and revision available today is probably free of errors and is easy to integrate with your system. Companies have, already, reported changes required to the standard which is therefore much more mature than it was in 2000. The ONIX standard became an international standard with the last version. This indicates that the standard is very good, and covers the need that companies all over the world have.

According to EDItEUR's newsletter from the last steering group meeting the 06th of October 2004 at the Frankfurt Book Fair (EDItEUR10 2004) there are several countries implementing the ONIX standard. Since this is an international standard this means that companies from different countries can exchange information about books to each other without any problems, which again makes it easier to import and export books.

APA (APA2) informed that it is very difficult getting the book information from the publisher to the booksellers without using a standard, since the companies use different format preferences. By using a standard format for exchanging the information make the publishers job less difficult and time consuming. Since it would be easier to exchange the information, I think the publisher would exchange more information than they do today. And the more information a book has stored, the easier is it to sell it.

For the publisher has the ONIX standard the following benefits (Lotz):

1. The publishers can control the book information sent to trading partners, particularly the online booksellers and managers of industry-wide databases
2. Make sure that the master databases of the industry have the right data about the products.

According to APA are there other benefits the publisher have of the ONIX standard (APA1 2003):

1. Editors can provide content-rich title information
2. Marketing can send better data to bibliographic agencies and improve promotional opportunities
3. Sales have access to accurate price & availability
4. Customer services save time dealing with queries
5. IT saves time spent on programming for different suppliers
6. Management save time and money overall

The implementation of ONIX has the following advantages for the book stores (APA1 2003):

1. Standardizes information supply from publishers
2. Uses an international standard of subject codes
3. Provides detailed price & availability
4. Provides richer information than BISAC eg website links, jackets, graphic details
5. Allows more time for book selection
6. Matches up Book track and publisher information
7. Helps build your websites
8. Improves your systems efficiency

Besides advantages for the publishers and book stores have the implementation of ONIX also advantages for the whole supply chain in the book industry (APA1 2003). The standard will provide a great resource for all marketing, website and information needs. Another advantage they mention is that it does not cost that much to implement since the standard is XML based and portable.

7.5 How ONIX is implemented in several countries

The ONIX standard been adopted by the following countries: US, UK, Australia, Canada, Republic of Korea, Germany and France. There are also several others countries who have started to use the standard. Below is there a description of how the implementation of the ONIX standard is coming along in several countries.

This information are based upon report from the steering group meeting the 15 of March 2004 at the London Book Fair (EDItEUR11 2004) and the last steering group meeting the 06 of October 2004 at the Frankfurt Book Fair (EDItEUR10 2004).

Australia:
In Australia there had been a rapid adoption of ONIX during a two-year period of government funding which is due to finish in June 2005. The Australian Publishers Association had established a national price and availability database, made extensive presentations, provided telephone and web support to users, and adopted a marketing approach to the selling of ONIX to publishers.

From the last report it was reported that publishers adopted the standard rapidly during the funding period. The book stores were also starting to take an interest in the standard.

Finland:
The first report informed that Finland were working on setting up a national database using ONIX, which was led by a steering group of booksellers who were demanding better price and availability information from publishers.

The last report informs that product data are now being transmitted using ONIX, and web-based input tools were going to be developed. Finland had some local ONIX requirement, like additional codes that they would like to have added in a future release.

France:
In the first report it was written that in France they were working on the translation, notably of the code list, and a conference planned in June 2004. Further informed the first report that there was still some reluctance among the big publishing companies, but they already had plans for co-operation with some publishers.

The last report informed that the latest transition of the ONIX documentation would be complete in October. It also reported that the conference in June was very successful.

Germany:
The first report from Germany told there was a need for more participants and additional promotion in order to get the industry to adopt ONIX.

The second report informs that ONIX was beginning to be adopted by some publishers. The translation of the ONIX documentation was also about to begin. The code lists were already translated.

Italy:
Italy is one of the latest countries who joined the ONIX International Steering Committee (Editeurnews 2005).

An ONIX translation is shortly finished according to the last report. The standard engaged the trade's interest.

Korea:
In Korea they have set up a new supply chain organization in June 2003 with the priority to promote use of ONIX. All systems were in place for ONIX to be used as a standard and as the basis for a national database, but publishers had not yet begun to implement it.

Netherlands:
In Netherlands ONIX had become more widely adopted by publishers according to the first report. The second report informed that Centraal Boekhuis had decided to adopt ONIX.

Poland:
In Poland where there is little standardization today, a standards committee has now been formed.

Norway:
In Norway there is some interest in ONIX from the publishers, but there has not been any initiative yet.

UK:
The report from the meeting in March informed about the following:
In the UK EDIFACT was widely established in the library sector and by Amazon.com. Nielsen BookData reported that they received 24 live ONIX feeds in addition to one from New Zealand. The next priority was to develop outbound messages. The ONIX group was now active

in development of the standard.

From the meeting in October the following was reported from UK:
The library sectors usage of EDIFACT had increased, whereas the wholesalers were the only sector who adopted XML.

BIC had a project called e4books to get greater adopting of e-commerce in the industry.
ONIX was growing in the UK.

US:
In the US more than 200 publishers, including 19 of the top 25, were now sends ONIX messages. This was informed in the report from March along with information that Ingram were taking the 2.1 versions and beginning an outbound ONIX feed.

The report from October informed that most major publishers are now sending ONIX messages. However they were interested in getting the smaller publishers to also adopt the standard.

7.6 How ONIX is used at Amazon

Meier wrote that Amazon started with Baker & Taylor as their single catalog source in 1995, a short time after, Ingram was added (Meier 2003). These leads to the need of compare the multiple data sources. Amazon learned the value of good quality data, since this would effect the number of sales that would be done, and they saw that some data sources were more reliable the others.

According to Publishing Technologies (PublishingTechnologies 2000) Amazon and Barnes & Noble were among the first companies who adopted the ONIX standard when it was introduced in 2000. Amazon started testing the standard with several large US publishers, to see how the standard worked, and to get them to adopt the standard as well. Amazon wanted all their partners to start using the ONIX standard when sending their book, music and video data.

In 2002 there were few publishers that had adopted the ONIX standard, which made the advantages for the companies that did less visible. Barnes & Noble tried to improve the situation by giving their publishers an ultimatum (PublishingTrends 2002), where they gave the publishers a 6 month limit for when they had to start sending information using the ONIX standard. If the publishers could not get this done, they would have to pay to get it done. The timeframe however was not definitive.

Amazon work today together with other major booksellers such as BN.com, Borders and Fathbrain on making ONIX the sole standard for transmitting information about books (BISG1).

From Amazon's website (Amazon) you can read how you can send files to Amazon, with book information:

Amazon wants publishers to send files via FTP to their catalog folder, or, if it is a major publisher who regularly uploads data, they want them to send the XML file to the catalog folder, and notify their regular catalog specialist. After publishers send the files via FTP, they will have to write to book-typos@amazon.com with the file names.

If publishers are unfamiliar with sending files via FTP they can read instructions on how this is done on this site.

I sent a mail to Amazon where I asked them if all companies in the book industry exchange information with them using ONIX. I also wanted to know if it was a difficult process to get them to use ONIX. I received a reply from them the 13th of October 2004 where they informed me that

not all publishers are sending information to them via ONIX. However they are still in the process of migrating publishers to the ONIX format, as they become prepared.

7.7 MARC and ONIX

MARC is a short term for Machine-Readable Cataloging, and is a data format that emerged around 30 years ago by the Library of Congress (Congress). MARC became USMARC in the 1980s, and in the late 1990s it became MARC 21.

MARC was designed to store bibliographic information in the library. ONIX will not replace MARC, but supplement it. ONIX will make it easier to transfer or display the information, which MARC is not that suited for (Dawson 2003).

The MARC formats are standards for the representation and communication of bibliographic and related information in machine-readable form.

(MARC1)

A MARC record contains information like description of an item, main entry and added entries, subject headings, classification or call number and some additional information. This is information that normally would be on a catalog card (MARC2).

It is possible to have a machine read and interpret the data in a cataloging record.

It became possible to map from ONIX to MARC21 from version 2.1 of ONIX.

Chapter 8
The XML standard

This chapter will give a description of Extensible Markup Language (XML) and some standards that XML has a close relationship to. The first standard I will describe is the Standard General Markup Language (SGML), since XML is a lighter version of this standard. I will for each standard explain how it is related to XML.

XML is as mentioned in the previous chapter used by ONIX when exchanging information between companies. XML is used to structure the information in order for both parties to know what the information is referring to. Later in this chapter there will be some illustrations which show the XML elements an ONIX Product Information Message consist of.

8.1 Standard General Markup Language (SGML)

SGML with ISO 8879 is an international standard for defining a description of the structure of different types of electronic document. The SGML standard is very large, complex and powerful and not well suited for serving documents over the web. XML was therefore created in the early 90's so that richly structured documents could be used over the web. XML is a lighter version of SGML, and consists of some of the functionality, enough to be a useful tool (Flynn 2004).

XML makes it possible to deliver “well-formed” data without a Document Type Definition (DTD) which was not possible with SGML. SGML requires that the structured documents must be “valid”, and have a reference to a DTD. I will give a more detailed description of “valid” and “well-formed” documents later in this chapter.

8.2 Extensible Markup Language (XML)

XML was designed with the intention of improving the functionality on the web by being more flexible and adaptable. XML makes it easy to transmit and share information across the web, and to store structured information. XML allows people or organizations to develop their own customized markup application for exchanging information in their domain. XML is therefore a kind of metalanguage, which is a language for describing other languages. The idea behind the design of XML was to make it easy to use SGML on the web. However, XML is used to store all kinds of structured information, not just web pages (Flynn 2004).

XML was developed by W3C Generic SGML Editorial Review Board formed by W3C Consortium and was chaired by Bosak from Sun Microsystems. The Generic SGML Working Group organized by W3C also participated. Today XML is supervised by the XML Working Group (CoverPages2 2004).

The syntax rules for XML are simple and very strict:
An XML document must have a root element and all the elements have to be properly nested together. It is also important to remember that all elements must have a closing tag and that the XML tags are case sensitive. The attribute values have to be enclosed in quotes, and there shall be no repeating attributes in an element (Pascall).

XML itself does not do much, but it helps us with structuring, storing and exchange of information. You can store the data in files or in databases, and use an application in order to store or retrieve information (W3Schools1).

As mentioned in the previous sections are there two forms for XML documents:

- Valid
- Well-formed

The difference between valid and well-formed XML documents can be explained with the following statement:

"Every valid document is well formed, but not every well formed document is valid. "
(HWG.org 1998)

A well formed XML document must follow three simple rules (HWG.org 1998):

1. It must contain a least one element
2. There must be a unique opening and closing tag, which contains the whole document (this forms the root element)
3. All the tags must be correctly nested and must match

Here comes an example of a well formed XML document:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Person>
  <Name>Hans Hansen</Name>
  <Born>01.01.2000</Born>
  <Address>
    <Street name>Fantasiveien 1</Street name>
    <Zip code>1000</Zip code>
    <City>Oslo</City>
  </Address>
</Person>
```

Figure 8.1: A well formed XML document

The well formed XML document above meets the 3 rules listed above. There are many elements in the XML documents, Person and Name to mention some of them. Person is the root element of the XML document, and we can see that the elements within them are correctly nested and match. The first line of the XML document identifies the XML version and encoding used in the document.

An illustration of how the elements in the well formed XML document above is connected:

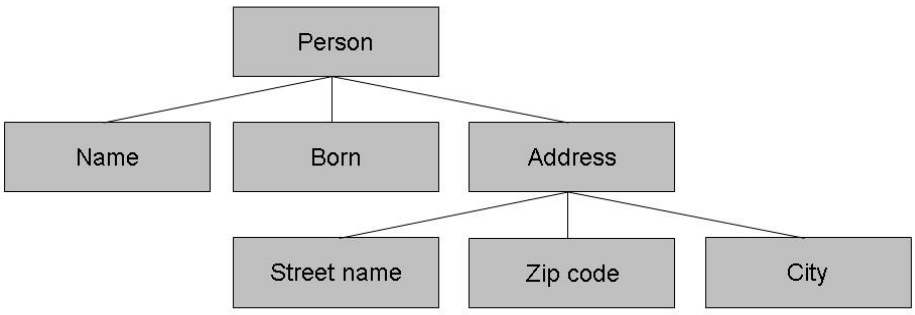

Figure 8.2: Illustration of a well formed XML document

A valid document must follow the rules that make it well formed and it must conform to its DTD, which tells how the documents shall be laid out (HWG.org 1998). In order to make the well formed XML document above valid, it must conform to a DTD. The illustration of the well formed and valid XML document would be the same.

Here is the well formed XML document above modified in order to make it valid:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Person SYSTEM "Person.dtd">
<Person>
  <Name>Hans Hansen</Name>
  <Born>01.01.2000</Born>
  <Address>
    <Street name>Fantasiveien 1</Street name>
    <Zip code>1000</Zip code>
    <City>Oslo</City>
  </Address>
</Person>
```

Figure 8.3: A valid XML document

To make the XML document valid the following line had to be added in the XML document:

<!DOCTYPE Person SYSTEM "Person.dtd">

The statement says that this is a document of the type Person, where Person is the root-node. The name of the DTD after SYSTEM informs where the DTD file, in this case Person.dtd, is located.

The DTD Person.dtd that Person.xml is referring to would look like this:

```
<!DOCTYPE Person [
<!ELEMENT Person (Name, Born, Address+)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Born (#PCDATA)>
<!ELEMENT Address (Street name, Zip code, City)>
<!ELEMENT Street name (#PCDATA)>
<!ELEMENT Zip code (#PCDATA)>
<!ELEMENT City (#PCDATA)>
]>
```

Figure 8.4: DTD

The first line of the Person.dtd shows that the element Person consists of three elements. The address element has a "+" after it is listed. That is because there could be several address elements. The second line shows the element Name is of type #PCDATA, which is the value.

A different alternative to describe the structure of an XML document besides DTD could be XML Schema language. I will give a short description of both in the following sections.

8.3 Document Type Definition (DTD)

A DTD defines the structure of a XML document. Information within it could be the name of elements, attributes and entities that can be used in a XML document. The DTD starts with identifying the root element of the XML document, and may also contain additional declarations.

You do not have to use DTD when you use XML, but by using it you will make sure that the information is structured correctly. A DTD can either be inside a XML document, or as an external reference. In the above example the DTD was an external reference. The DTD defines which elements are mandatory and which are optional in a XML document, and how many times an element will repeat.

Advantages by using a DTD are that you can validate your own data, you can use it in applications to make sure information people send are valid, or a group of people can use a

common DTD to make sure that they know what the information others send are (w3Schools2).

However, a big disadvantage with DTD is that it only specifies the structure of a XML document. It was designed to be used with traditional text documents, and not for rectangular or tabular data. With DTD you will not be able to specify numeric ranges or define limitations or checks on the data content. For this kind of operations XSD is better suited (Flynn 2004).

8.4 XML Schema Definition (XSD)

XML Schema Language is referred to as XSD. XSD provides richer support for XML than DTD, and will according to W3C probably replace DTD in most web applications in the future. An Advantage with XSD is that it is written in XML, supports namespaces and data types. XSD is therefore the recommendation of W3C for structuring XML (W3Schools3).

By using CSS or XSL it will be possible to show the information from the XML. I will give a short description on both below:

8.5 Cascading Style Sheets (CSS)

CSS makes it possible to display the information listed in a XML documents. W3C does not recommend using CSS to display information on the web, since XSL is not supported by the main browsers yet, but this is the only tool available (W3Schools4).

8.6 eXtensible Stylesheet Language (XSL)

XSL is recommended by W3C to use when you shall display information from an XML document on the web. XSL transform XML into HTML before it gets displayed in the browser. Not all browsers support XSL yet, but today both Netscape 6 and Internet Explorer 5 support it (W3Schools5).

8.7 HyperText Markup Language (HTML)

HTML is a small application of SGML used on the web, which defines very simple classes of report-style documents. While HTML is a predefined markup language with fixed format, XML does not have a fixed format, but is a “metalanguage” where you define your own tags and the document structure. With XML you can also define your own markup language for other documents (Flynn 2004).

XML is not a replacement for HTML, since they are designed with different goals. XML is designed to describe data and focus on what data is, while HTML is designed to display data and focus on how the data looks.

I will use the rest of this chapter to explain and illustrate the ONIX Product Information message's relation to XML.

8.8 ONIX Product Information Message

The international ONIX standard bases their ONIX Product Information Message (ONIX message) upon XML. We can read from EDItEUR's website (EDItEUR12) why they chose to use XML in the ONIX messages:

- XML is optimized for creating complex documents and transmitting and exchanging data between computers.
- XML is text-readable, meaning that humans as well as computers can recognize and read the data. Most tags, which define each book data element, consist of English words or abbreviations--for instance, an ONIX message would list the Publisher's name as follows: "<PublisherName>Scribner's</PublisherName>". These factors make it easier for smaller organizations to design and implement ONIX-compliant systems.
- XML software is inexpensive, meaning that even smaller publishers can use it, which was a major goal of the ONIX committee.

Here is another statement explaining the advantages with using XML in combination with the ONIX standard:

"XML in general and ONIX in particular are data formats for text files intended to be generated by one computer program and parsed by another computer program. These text files are relatively easy for computers to generate and parse, and the XML data format is not specific to any programming language or operating system. All modern programming languages include facilities for handling text files, and most languages now include XML generation and parsing capabilities."

(Abiblion)

The DTD that the ONIX message refer to, consists of a number of linked text files, which together allow standard XML software to parse, verify and operate the content of the ONIX product information.

Instead of looking more at why XML was chosen to be used as the foundation of the ONIX standard, I will explore what the ONIX message looks like and understand how it is built.

An ONIX message consists of 4 parts (EDItEUR13):

1. the start of the message
2. the message header block
3. the body of the message
4. the end of the message

In the following sections I will explain and illustrate each part of the ONIX message. The information about the 4 parts written below is based on the XML message specification for the ONIX standard (EDItEUR13).

1) The start of the message
The start of the message has both XML format and content. A message starts with the following 4 lines:

```
<?xml version="1.0"?>
<!DOCTYPE ONIXMessage SYSTEM
"http://www.editeur.org/onix/2.1/reference/onix-international.dtd">
<ONIXMessage>
```

Figure 8.5: The start of an ONIX message (EDItEUR13)

The first line identifies that the document is according to the XML standard, version 1.0.
The second and third line makes the DTD declaration, which tells the XML software that the document is based on Release 2.1 of the ONIX XML DTD using reference names. The root element is defined; ONIXMessage. You can refer to either the “reference names” or the “short names” in the DTD. The URI tells us where we can find the files that correspond to the reference names of release 2.1 DTD, which is at EDItEUR's website. The DTD file is called onix-international.dtd, and it contains information about how the structure of an ONIX Product Information Message shall be, and which elements it shall contain. By comparing the XML document with the DTD file, you make sure that the ONIX Product Information Message is correctly set up.

The fourth line informs that the ONIX Production Information Message begins.

2) The message header block
The message header block contains data elements which specify who the sender is and the date of the message (mandatory), the addresses (optional) and default values for measure units, currency, etc (optional). You can place the elements inside a <Header> composite to make it easier for the companies, since they use the same composite for every message. It is not mandatory to send the data elements as a composite. They can also be sent as several header elements.

The following illustration show the first two parts of an ONIX message:

```
<?xml version="1.0"?>
<!DOCTYPE ONIXMessage SYSTEM
 "http://www.editeur.org/onix/2.1/reference/onix-international.dtd">
<ONIXMessage>
<Header>
<FromCompany>Portadas.net</FromCompany>
<FromPerson>Bernie Rabow bernie.rabow@portadas.net</FromPerson>
<ToCompany>EDItEUR</ToCompany>
<ToPerson>David Martin</ToPerson>
<MessageNumber>1213</MessageNumber>
<SentDate>200007311330</SentDate>
<MessageNote>Prueba en Espa&ntilde;ol</MessageNote>
<DefaultLanguageOfText>spa</DefaultLanguageOfText>
</Header>
```

Figure 8.6: An ONIX message header (EDItEUR13)

Looking at the data elements inside <Header> and </Header>, we can see that it is information about where the message is coming from, where the message is going, sending date and some other information. As mentioned earlier the mandatory fields in the header composite <FromCompany> and <SentDate>.

3) The body of the message
The body of an ONIX message can consist of the following record types:

- Product
- Main Series
- Subseries

There can be one or more records in the body of the message, in any mix. A single ONIX message can for example contain information about 2 Products and 1 Main series.

In Appendix A an example of a whole ONIX message is illustrated. The ONIX message contains one product record.

4) The end of the message
This part only consists of the statement: </ONIXMessage> which ends the ONIX message.

I will now show some diagrams from different parts of the ONIX Product Information Message. In the diagram symbols are used, which are not intuitive. I will therefore give a short description of the most important ones. The information about the diagrams, the diagrams and the symbols in this section are from the Intro file (Intro). The symbols inform how the connection between the elements to the right of an element is. Where there is no symbol between two elements in the diagrams, the structure is mandatory:

"?" "Optional and non-repeatable"
"1" "mandatory and non-repeatable",
"*" "optional and repeatable"
"+" "mandatory and repeatable".

If there is a square bracket between the first and last element to the right side of an element, then the structure is a sequence of child elements (elements to the right)

If there is an angle bracket between the elements to the right of an element, means that you will have to choose element to include.

The first diagram shows the top level of what an ONIX Product Information Message looks like (Intro). The ONIX message element is on the left side of the diagram. The ONIX message consists of the elements that are placed to the right side of it. From the diagram below we can read that the ONIX message must consist of either a sequence of header data elements, or the header data enclosed in a header element. The ONIX message must also consist of one or more Products, Main Series or Subseries records.

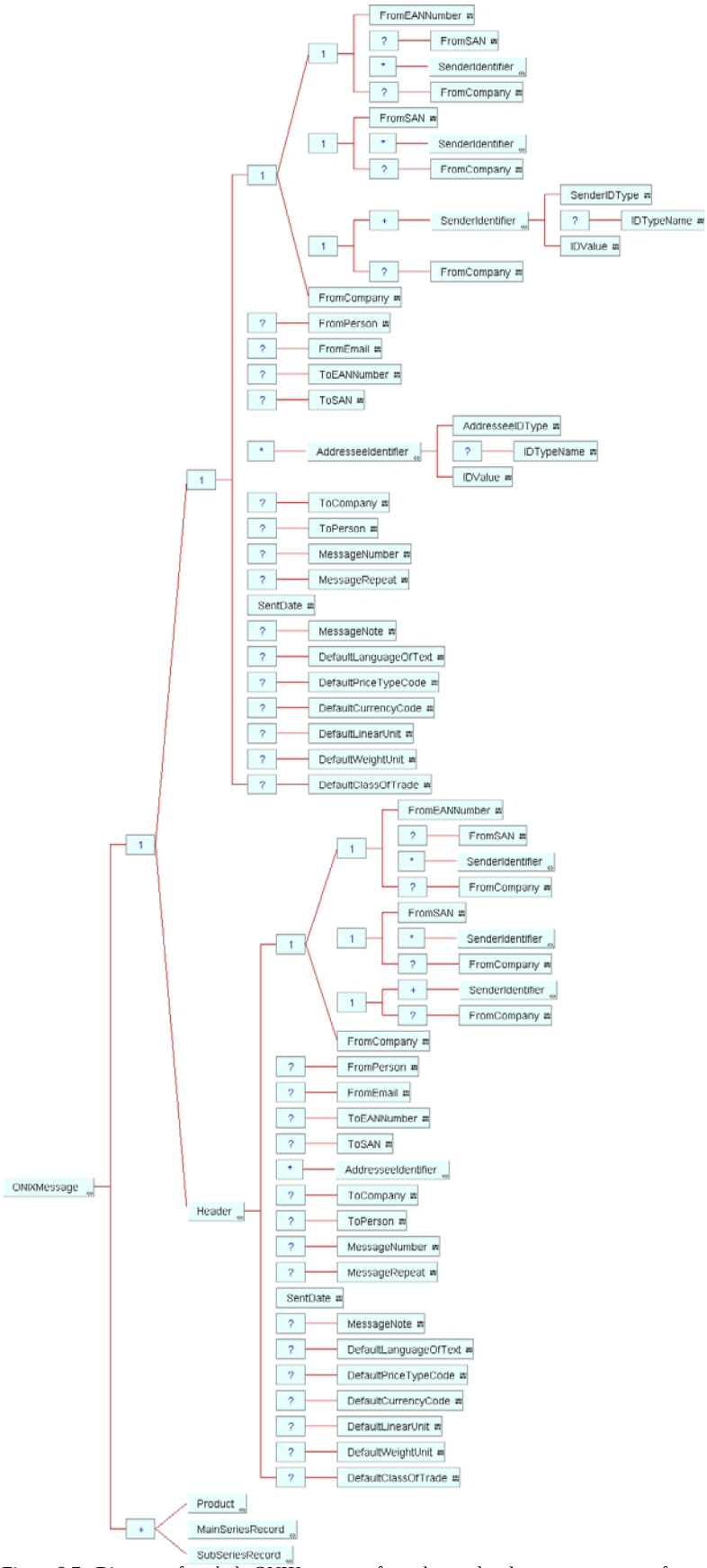

Figure 8.7: Diagram of a whole ONIX message from the top-level message structure from the Intro file (Intro)

Exploring the Product record shown at the top level from figure 8.7, we can see in the figure below that it consists of the following groups. The diagram on the left illustrate how the groups are connected within the Product record and the diagram on the right inform what the groups are called.

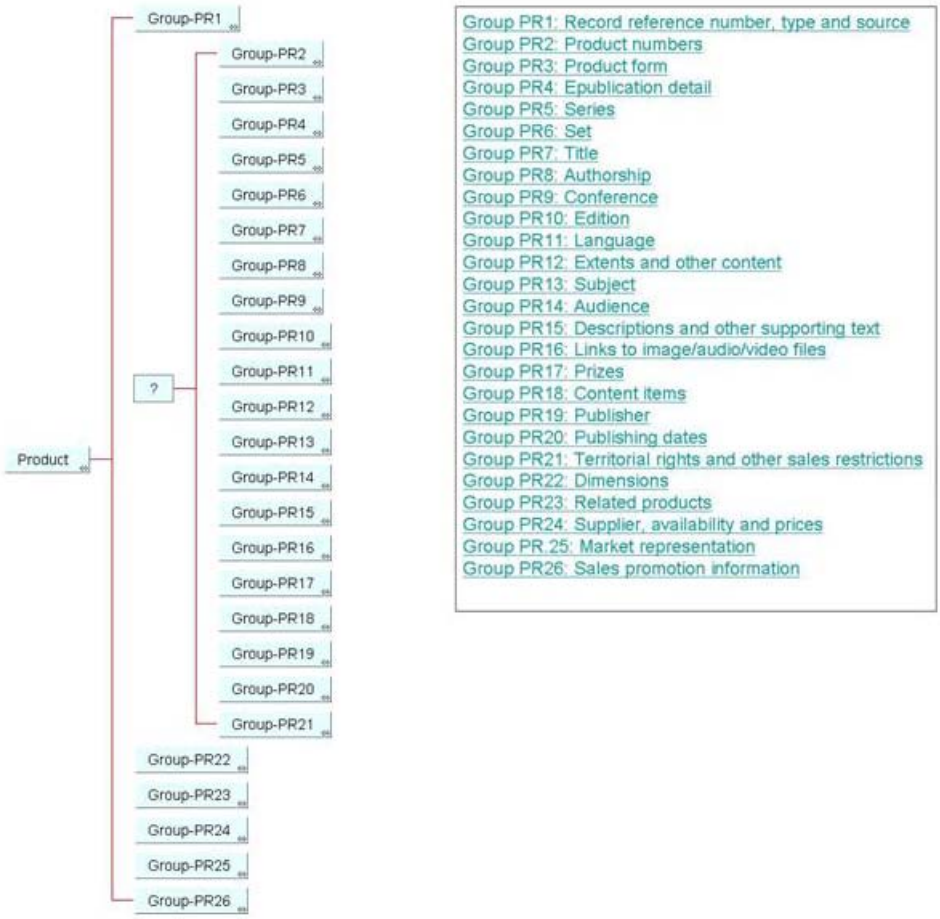

Figure 8.8: Diagram of an ONIX Product record structure from the Intro file (Intro)

The second diagram shows that a Product record has only one mandatory group, which is Group PR1 (Record reference number, type and source). The rest of the groups are optional. Groups PR2 through PR21 are basic bibliographic and descriptive data.

The diagram below illustrates which elements PR1 (Record reference number, type and source) from the Product record consists of. Record Reference and Notification Type are as mentioned above the only mandatory fields in this group. The other elements are optional and non-repeating.

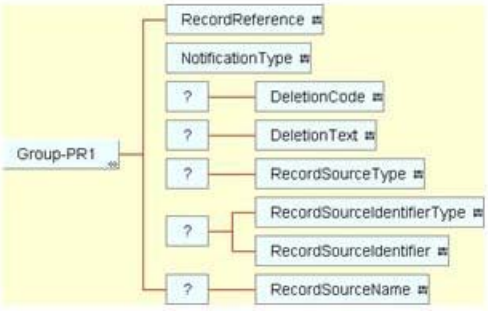

Figure 8.9: Diagram of PR1 in an ONIX message from the Intro file (Intro)

Chapter 9
How XML is supported in databases

In this chapter I will explore how several database solutions support storing of XML documents. I explore which database solution is best fitted for storing XML based ONIX messages, described in the previous two chapters.

I will begin this chapter by looking at the features important for the database to support. Then I will establish what kind of document type an ONIX message is, and what type of data it contains.

After this has been determined several appropriate database solutions will be presented and discussed.

A database is characterized as a place where data is stored, usually for a long period of time (Elmasri and Navathe 2003). By data he means known facts that have an implicit meaning. A database can be of any size, and it can have different degrees of complexity. How a database is designed depends on the purpose of the database, and what the database are used for.

A DBMS is a powerful tool for creating and managing data efficiently and allowing it to persist over a long period of time.

(Garcia-Molina, Ullman et al. 2002)

A database management system (DBMS) consists of several programs, and makes it possible for a user to create and maintain a database.

Three capabilities that a DBMS provides the users with (Garcia-Molina, Ullman et al. 2002):

1. Persistent storage: The DBMS allow for large amount of data to be stored, and the data exists independent of any process that is using the data. The DBMS also provide flexibility, because data structure can efficiently access large amounts of data.
2. Programming interface: The DBMS allow users or application system to access and modify the data by using a query language.
3. Transaction management: The DBMS allow concurrent access to the data, by having many small independent processes.

9.1 Features a database have to support

In order to find the best suited database solution for storing the ONIX messages, I have designed four features it is important that the database solution support. Later in this chapter I will discuss how the database solutions support these features:

1. Insert of the ONIX-message or data from the ONIX-message into the database
2. Retrieval of the ONIX-message from the database
3. Retrieval of data from an ONIX-message based on criteria in the ONIX-message
4. Update of data in an ONIX-message

In the XML chapter I wrote that an ONIX-message consists of both mandatory and optional elements, and several elements are also repeatable. A database solution therefore supports any type of ONIX-message that comes in, as long as it follows the rules in the DTD.

9.2 XML document type

The first step towards finding the right database solution is to determine what type of XML document the ONIX message are. There exist three types of XML documents (Elmasri and Navathe 2003):

- Data centric
- Document centric
- Hybrid

A hybrid XML document may contain some structured data and some predominantly textual or unstructured (Elmasri and Navathe 2003).

The following description of data centric and document centric XML documents: In the data centric document type you are interested in saving the data in the XML document, while in the document centric type you want to save the whole document. Typical examples of data centric documents can be sales orders or flight schedules, where they use only XML to transport the data. The order of the information does not matter, so there is no problem saving it in different tables in a relational database for example. Examples of document centric XML documents can be email and books, which do not originate in a database. The many small data items in a data-centric XML document follow a specific structure, while in the document centric the data is large and the data elements has little or no structure. If the XML document is data centric, the data could be used in data processing (Bourret 1999-2004).

The table below lists the main characteristics of the two XML document types:

Data Centric	***Document Centric***
Fine-grained data	Large-grained data
Order of element not significant	Order of element is significant
Machine consumption	Human consumption

Table 1: Characteristics of the two XML document types (Bourret 1999-2004)

As a general rule data centric documents are stored in traditional databases that are relational, object-oriented or hierarchical, while document centric document shall be stored in a native XML database or a content management system. However, this rule is not definitive, which means that the data centric document can also be stored in native databases and visa versa (Bourret 1999-2004).

Based upon the above description of the three document types, an ONIX-message can be viewed as a data centric document. The reason for this is that it is the data inside ONIX-messages that are of interest, the data about a book. XML is only used to transport the data from one place to another. Other factors that point towards data centric are that the data within the ONIX-message are fine-grained, and the order of the elements is not significant.

Even if most factors point towards a data centric document, there is also one factor that indicate that the ONIX-message are a document centric document. This factor is that it have to be possible to retrieve the whole ONIX-message, and for it to look like the original one. This is necessary if you for example want to transfer the ONIX-message with all the information to someone, and to look at all the information stored about a product.

9.3 XML data type

An XML document contains three different types of data (Elmasri and Navathe 2003):

1. Structured data
2. Semi structured data
3. Unstructured data

Structured data has their representation in a strict format. The data are stored in a database, and all the records within it follow the same format. Semi structured data is data collected before knowing how to store it. Since the data is collected ad hoc, they do not have the same structure. By looking at the XML document it is possible to determine whether the data is structured or semi structured. If the XML document conforms to a predefined XML scheme or DTD, it is structured. If the XML does not conform to any XML scheme or DTD, it is semi structured. Unstructured data is when there is little indication of what type of data it is (Elmasri and Navathe 2003).

The ONIX-message illustrated in Appendix A conforms to the following DTD-file: http://www.editeur.org/onix/2.1/reference/onix-international.dtd.

The ONIX-messages are therefore structured.

9.4 How XML and ONIX is supported by several database solutions

In this section several database solutions are presented and how they support the features defined earlier in the chapter. The information about the databases are mostly from the XML and Databases article written by Bourret (Bourret 1999-2004).

The database solutions described below are divided within the two document types; data centric and document centric. Even if I determined earlier that the ONIX-message could be viewed as data centric, I have included both categories here because the ONIX-message had some factors that indicated it to be document centric.

9.4.1 Data centric databases

Several databases could be used to store data centric documents. Relational databases are used when storing data from XML documents, so I will therefore give a more detailed description of this type of database.

Relational database

A relational database stores information in rows and columns in tables. The relations between the tables are supported by foreign keys, which link the information in the tables together.

A relational database could be used to store the data from a XML document, but if this is an appropriate solution or not depends on the complexity and size of the XML document. If the XML document is small and simple, a relational database is a good alternative. If the XML document is of a larger size and more complex, a relational database would have many tables, which would make insert and retrieval of XML document more difficult.

The onix-international.dtd (EDItEUR7 2004) informs that an ONIX-message can be large, and very complex. This is because an ONIX-message has both mandatory and optional elements, and the elements are repeatable. A database would have to be set up according to the DTD, to make sure it would support all the different kinds of ONIX-messages that arrive.

Storing ONIX messages in a relational database would require many tables to make sure that all the information is supported, and to avoid redundancy. Since many of the elements are optional

and not important for the companies in the book industry, many of the fields within the tables will be empty. To have many empty fields within tables or many tables in total is not a well suited solution.

XML is a hierarchical document that can be translated into normalized relations. Normalizing the XML structure into relations in a relational database could be a complex task for the designers, time consuming for the programmers and operationally inefficient for the users and database administrators. (Champion 2001)

The following figure illustrates the mismatch between XML data and Relational databases:

XML	***RDBMS (Normalized)***
- data in hierarchical structure	- Data in multiple tables
- Nodes have element and/or attribute values	- Cells have a single value
- Elements can be nested	- Atomic cell values
- Elements are ordered	- Row/column order not defined
- Schema optional	- Schema required
- Direct storage/retrieval of simple docs	- Joins necessary to receive simple docs
- Query with XML standards	- Query with SQL retrofitted to XML

Table 2: Mismatch between XML data and Relational databases (Champion 2001)

A short description of how the relational database supports the 4 features listed in the features section:

- It is a very time consuming and complex task to normalize the XML document into rows, columns and tables, and to set the relationship between them, especially if the XML document is large and complex so there have to be many tables.
- Searching for the whole XML document in a relational database will be slow if the database consists of many tables. That is because there would need to be many joins between the tables, which make the search inefficient.
- Searching for data from parts of the XML document is very fast, even if there are many tables. This is because then you know which table the information is in, and you do not have to get out the whole document.
- Update of data is fast since the data from a document are stored in tables, and it is easy to navigate to the table where the information is.

To make it clearer why relational databases must not be used to store complex XML documents like the ONIX messages, will now be illustrated. The first XML document is a small and simple one, while the second one is larger and more complex. One of the reasons for why it is more difficult to store complex XML documents in a relational database is because XML uses a hierarchal (tree) model, while the relational data model is flat.

The small and simple XML document:

```
<Person>
  <Name>Hans Hansen</Name>
  <Born>01.01.1970</Born>
  <Home_number>12345678</Home_number>
  <Home_number>12345679</Home_number>
<Person>
```

Figure 9.1: A simple XML document

The XML document above would result in two relational tables (Person, Home_number). The Person table contains the primary key when the person was born, and the name of the person. The Home_number table would contain information about the telephone numbers the person have, and the primary key from the first table, when the person was born, to connect the tables together.

The larger and more complex XML document:

```
<Person>
  <Name>Hans Hansen</Name>
  <Born>01.01.1970</Born>
    <Children>
      <Child>
          <Childs_name>Kari Hansen</Childs_name>
          <Favorite_toy>Doll</Favorite_toy>
      </Child>
      <Child>
<Childs_name>Finn Hansen</Childs_name>
          <Favorite_toy>Football</Favorite_toy>
          <Favorite_toy>Bike</Favorite_toy>
       </Child>
    </Children>
  <Home_number>12345678</Home_number>
  <Home_number>12345679</Home_number>
  <Cell_number>87654321</Cell_number>
<Person>
```

Figure 9.2: A more difficult XML document

The XML document above is a more complex version of the one from the first example, and would require 4 tables (Person, Childs_name, Favorite_toy and Home_number). Person and Home_number would be the same as in the simple example, while Childs_name would contain the name of the child and the primary key from the Person table, and Favorite_toy would contain all the toys the child has, and primary key in Childs_name, which is the childs_name.

The illustrations show that a relational database could be used if the XML document is small and simple, while it is difficult to use if the XML document is large and complex. The complex illustrations contained around 10 elements, and required 4 tables. An ONIX message consists of over 250 elements, and would therefore contain many tables. From the more complicated example you can see that some of the elements are repeatable, which lead to extra tables. If the element is repeatable, it would have to be stored in a separate table. From the onix-international.dtd (EDItEUR7 2004) you can see that many of the elements for ONIX is repeatable, which again leads to even more tables.

Object database

An object database consists of objects with data and behavior. Querying objects are often done with help of methods because the object database contains a variety of attributes and data. The data modeling for object databases are similar to the data model for XML, since both has an object or root, it contains nodes, and the nodes can contain data.

An object-oriented database is according to a Chaudhri (Chaudhri) an overkill for structured text. Since XML is not object-oriented, and have no inheritance, encapsulation nor behavior.

Object-relational database

An object-relational database is, according to an Champion (Champion 2001) a relational database with modifications. There have been added some features for making the XML data management easier, especially when it comes to serializing the data from object-oriented programs and databases. An important change in the relational database is that they have added Large Object (LOB) data types, which let data be stored and retrieved in a single cell of a table. The features added to the relational database make it easy to store and retrieve XML data.

Hierarchical database
A hierarchical database is a tree which contains data nodes (Mertz 2001). The data nodes can again contain data or other nodes. An advantage with hierarchical databases is that it does not require deconstruction and reconstruction of the XML structure if you want to get data in or out of the database. Retrieval and update of XML data are also highly optimized.

The hierarchical database is fast on location of single nodes and sequences of nodes, but slow to enforce the structure of XML. It is also difficult to search within an unknown structure of elements within a hierarchical database, which is necessary in a database.

9.4.2 Document centric databases

There are several databases that belong to the document centric category, and the native database is a very good alternative for storing XML documents. I will give a short description of some of the alternative document centric databases, and a more detailed description of the native database, since this is a good database solution for XML data.

Flat file
To store XML documents in flat files means that each XML document will be stored as a file. The loading and storing of the documents will happen fast, while the searching will be very slow. File systems work best if there is a simple and small set of documents (Bourret 1999-2004).

To store the ONIX-messages in flat files is not a good solution, because you must be able to search for data or documents, and this is not supported in flat files. Another reason is that ONIX-messages are large and complex, which makes files inappropriate.

Native XML database
Native XML database is probably the best alternative when it comes to storing of XML documents that can be viewed as both data centric and document centric (Bourret 1999-2004). Native XML documents are especially good for semi-structured data, which are data in a regular structure, but the data amount varies because some of the elements are optional. Using relational database structure with this kind of data would as mentioned above result in a large number of columns with null values (waste of space) or a large number of tables (inefficient to retrieve) when mapping.

Indexing an XML document is possible, but it makes the searching of a XML document a bit more difficult (Williams 2001). It is also possible to decide which element you want indexed.

The native XML database stores the whole XML document.

The native XML database supports the 4 features as following:

- The XML documents are easy to store using native XML database.
- The retrieval speed of data of a XML document is high, compared with how long this takes with relational databases. This is because the entire XML document is stored together physically, which means that you do not need joins
- Queries on data between several XML documents may on the other hand be slower in native XML databases than in relational databases. This also depends on how complex the database is.
- Slow update is a problem in native XML databases.

A description of two forms of native XML databases (Bourret 1999-2004):

- Text based where the XML is stored as text. This can for example be a file in a file system or a BLOB in a relational database. The text based native XML database has indexes, which lets the query engine jump to a specific place in the XML document. This indexing lead to a speed advantage when retrieving documents or fragments of documents
- Model based where they built an internal object model from the document, and store this model. They can store the model in a relational or object-oriented database, depending on the database.

By using a native XML database you may retrieve the original XML document, which is something that is important when it comes to the ONIX-messages.

BLOB (Binary Large Object)
BLOB as a data type used to store XML data. It is related to the relational database, because you can store the XML document or parts of it in tables inside a relational database. When storing XML documents as BLOB, you can make your own XML index table which will contain the index value and a foreign key, which points to the primary key in the document table. Besides the primary key, the document table contains the XML document stored as a BLOB (Bourret 1999-2004).

BLOB is fast on insert and select for XML documents, but slow when it comes to queries (Suleman 2004).

However, BLOB is well suited for semi-structured data, and queries become more efficient if you choose to index the XML document (Chaudhri).

9.5 A report that discuss three XML database solutions

CINCOM, which is an experienced software company, has written a white paper (CINCOM) where they compare the three following XML DBMS:

- Relational
- Text-based
- Object-based

CINCOM thinks that these DBMS are better then other solutions within XML applications, because they improve productivity and reduce costs. They think Object-based systems are the best since they have the right characteristics to support high performance and robust applications. Below is a summary of how they see the databases within the three categories:

Relational-Based Solution:
Relational database management tool is typically used when new technology is emerging. The XML is mapped to relational schema, which is easy if it is a strictly hierarchical data model. As the document becomes more complex, the data model will contain more hierarchical levels, and there will be a need for many foreign keys. This database management tool does not support storing of white space, and comments, DTD and process instructions are dropped.

We can see that the structure of a relational database is different from the structure of an XML document. The relational database consists of rows and columns with relationship modeled with foreign keys, while the XML document has a hierarchical data model. A relational database is therefore not the preferable when the XML document does not have a simple structure.

From the table below we can see that a relational-based solution is the worst alternative, because many of the limitations that XML provides are not supported. Relational databases are only used

for data-centric XML documents, where you only need a couple of classes. For more complex XML documents it is not efficient to use relational databases.

Text-Based Solution:
Text-based solutions are when XML DBMSs store XML as text strings. The text can also be indexed, to speed searches. The text-based solutions can be stand-alone or stored as a BLOB in a relational system.

Native XML database is a text-based solution where the XML document is stored as a text string. This solution has very fast storage and retrieval of XML documents, since it only require copy of the string to or from the database. The text can be indexed, so the queries are faster. Maintaining XML documents are difficult. First you have to copy the string into memory, and then find where you are going to change the data. If new data has a different size, there has to be support for this since the memory buffer and data will have different size. When sending the string back to the database, the new string must replace the old one and the string has to be reindexed.

Text solutions has ok performance when the data are mostly text, and there is little processing and updating.

Storing BLOB in a relational system has the same advantages and disadvantages as storing text-based implementations as stand-alone databases. However, by using BLOB in a relational system only use the database as a recoverable string file system, incurring all the overhead of the relational implementation without the benefits. The BLOB in a relational system is therefore maybe not to be recommended compared with a stand-alone solution. However, other considerations like IT-operational aspects may also come in. If for example there is a suitable database for other purposes, it might be smart to use this.

From the table below we can see that text-based solutions support all the XML, but because of lack in performance where data must be reparsed, wasting CPU and memory, and weak update capabilities the solution is not the best.

Object-Based Solution:
Object-based solutions store the different parts of the XML document as objects in a database. An object can for example be an element in the XML document, and the attribute of the element becomes an attribute for the object. The relationship between elements, their sequence and position in the hierarchy are supported.

Object-based solutions provide good processing characteristics in both text-related and non-text-related situations because they access the elements directly.

Compared with text-based solutions, there is more work inserting the XML document into the database, because the XML document has to be decomposed into objects with attributes. Reads and updates are on the other hand very fast. On reads, only the part that you want to read are copied into memory and on updates only the data that need to be updated are affected.

From the table below we can see that object-based solutions are the best alternative when storing XML documents. The solution provides efficient data update and access, including fragment processing capabilities and full support for XML syntax.

Criterion	*Relational*	*Text*	*Object*
Ability to treat XML as data	Yes	No	Yes
Support all XML syntax	No	Yes	Yes
Document / Fragment manipulation	Some	May be low performance	Yes
Search XML documents	Inefficient	Efficient	Efficient
Support XML-based APIs	Inefficient	Inefficient	Efficient
Retain all content	No	Yes	Yes
Round-tripping	No	Yes	Yes
Legacy data integration	Yes	No	Yes

Table 3: How features for storing XML documents are supported

9.6 EDItEUR's thoughts regarding a database solution

I sent a mail to one of the contact persons at EDItEUR, to get his opinion about which database solution he thinks is the best for storing ONIX data. I told the person from EDItEUR that the three main alternatives are a relational database, an object-oriented database or a native XML database. The person from EDItEUR told me that all three database solutions are being adopted today since it is easy to import XML data into a database. The contact person forwarded my mail to a technical consultant from the ONIX support team at EDItEUR, since he might have some comments about the native XML database.

The technical consultant told me that larger publishers use a relational database to store the ONIX data, because this is the most widely-used database technology. To store the ONIX data in a relational database is not straight-forward since there are over 250 ONIX elements. Most users do not need all the ONIX elements, so they do not implement them either.

Regarding the object-oriented database, the technical consultant told me that it supports inheritance and other features which are not necessary for the support of XML based ONIX messages.

The technical consultant was not fond of the use of XML database technology to support ONIX data either, since most of them is based on document content applications.

After some emailing, discussing which database solution that would be most appropriate, we agreed that a hybrid database solution is best suited to support ONIX data, since the data have to be queried fast and the ONIX message have to be retrieved as well. A hybrid database solution is a combination of a relational database and the use of BLOB. The most important information could be stored in relational tables, while the information that is not important could be stored in BLOBs. The information stored in the relational tables is typically information that you are interested in searching after or updating. The BLOB contains information that is usually not asked for, since searching and updating in BLOBs is a difficult task. Storing information in BLOBs, instead of having only relational tables, will make the retrieval of the ONIX messages much faster. The amount of tables necessary depends on which information the book industry value. The information the book industry does not think is important could be stored in BLOBs. The

insert of the ONIX-message will also be much faster in a hybrid solution than in a pure relational solution, since there will not be as many tables.

I discussed the hybrid database solution with an IT Manager from the book industry, and he suggested some modifications to the hybrid database model. He thought that there should be three tables instead of having two. The third table should be a Dictionary table, where information neither important nor non-important is stored. The information stored in this table are somewhere in between. The users are able to search for information stored in this table. The type of information stored in this table is usually optional, and therefore not always present. Having a separate table for this kind of information would decrease the empty fields within the tables with important information, and it would also make some less important information available for searching.

There is no right or wrong database solution for storing XML documents. It is important to choose a database solution that supports the XML document you want to store. Based upon the information about the different database solutions, I choose to use a hybrid database solution where I combine the relational database solution and the use of BLOB.

The next chapter will look at how the hybrid database solution with the three table types could be used to store the information from an ONIX-message.

Chapter 10
A hybrid database model

In the previous chapter I concluded that a hybrid database is the best solution for storing the XML based ONIX messages (EDItEUR6 2003). As I wrote in the introduction for the ONIX chapter are some of the documents referred to from the ONIX standard downloaded from EDItEUR's website. In this chapter I will look at what the hybrid database could look like, and how an ONIX-message could be stored within it. The hybrid database is designed with regards to what kind of information 2 book store chains find important.

10.1 A hybrid database model

A hybrid database is in this setting a combination of a relational database, storing the XML as BLOB in the BLOB table, and storing information in a Dictionary table.

Since there are many elements within an ONIX-message that are optional and/or repeatable, it is a good idea to divide the information in three categories, and store the information in different ways. The three categories the elements within an ONIX-message belong to are the following:

- Important for commercial use (Business transactions)
- Less important since it is not needed in commercial transactions, but wanted for informational purposes.
- Not important

Important information is information of very high interest among the parties. Examples of important information are ISBN, title, author and price. To make the information easily accessible for the companies when they are searching for it, the information is stored in relational tables like you do in a pure relational database model.

The less important information is information that is not of great interest among the parties. The information is therefore put in a common Dictionary table, where you can easily search for the information and update it. To insert the information in a common Dictionary table would take some time compared to storing BLOB in a table, since the information is not stored directly as XML data, but as text. One of the reasons for storing the less important information in a common Dictionary table instead of making more relational tables is because this type of information is often not present in the ONIX messages. By using a common Dictionary table to store the less important information in an ONIX message would reduce the amount of white space in the relational tables if the information were meant to be stored there. If for example one type of information would appear in an ONIX-message 1 time out of 100, it would be better to store the information as 1 row in the Dictionary table when it appears instead of having 1 non-empty field and 99 empty fields in a relational table.

Information within the not important category is information that are of no direct business interest for the parties. The information are almost never searched for nor updated, but only asked for when somebody tries to retrieve the whole ONIX message. The XML data from the ONIX message are stored directly as a BLOB in a common BLOB table, which makes insertion and retrieval very efficient. On the other hand searching and updating among XML data in a BLOB is complicated, and inefficient. BLOB is used if the information is of no importance, and you have to have a place to store it in case it would appear in an incoming message. By using a BLOB to store this type of information instead of having relational tables where this information could be stored leads to fewer relational tables. There is no point in having many relational tables with

information of no importance.

Below is a figure of how the database model would look like. I have only included some tables to give an impression of how the different table types are combined. In the big circle are all the relational tables connected, while in the two small circles are the Dictionary table and the BLOB table. All the tables are connected.

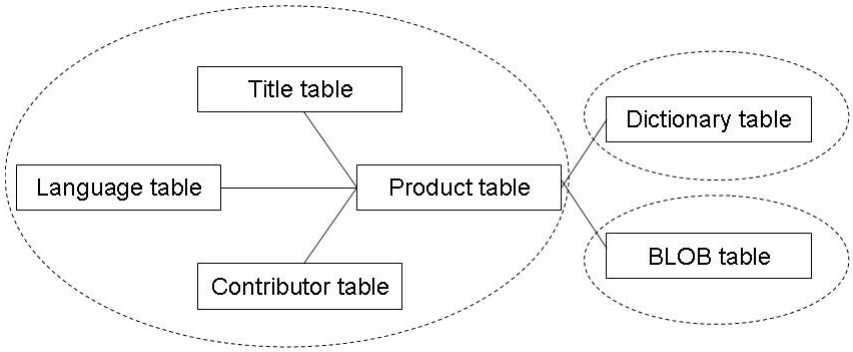

Figure 10.1: A small illustration of the hybrid database model

10.2 How to place an ONIX message in a hybrid database model

An ONIX Message contains a header item, and either one or several Product, Main-series and/or Sub-series items. The header composite is a small composite with X elements, where only two elements are mandatory; sentDate and fromCompany. The composite also includes a primary key that is set automatically. The two mandatory elements and the primary key are saved in a Header table, while the remaining optional and not so important elements are stored as a BLOB in the BLOB table. The reason for this is to prevent the header table from containing a lot of white space, since it is only sentDate and fromCompany that is mandatory. It is not a problem to extend the header table if you want to include some additional elements in to the table.

In appendix A there is an example of an ONIX Product Record Message. This ONIX message will be used in this chapter to illustrate how an ONIX message is stored in a hybrid database solution. The header composite of the ONIX message in appendix A consists of the two elements sentDate and fromCompany. There is no primary key visible, so the column in the table remains empty for now. However, in the real world the auto number can not be null since this is the primary key.

The Header table would look like:

autoNumber	**sentDate**	**FromCompany**
	20031215	Cappelen.no

The Product, Main-series and Sub-series are built up in a similar way, but in a much larger and more complex manner. I have only looked at how the Product Record (PR) is stored in the hybrid database model, since this is the record for the product itself. Main-series and Sub-series are additional records to the Product, if a book for example is within a series.

From the Product Record Format for ONIX for Books (EDItEUR6 2003) it is written about all the elements an ONIX Product Record consists of. The elements within a Product Records (PR) are divided into 25 different groups, where PR.1 is the only mandatory one to be included in an ONIX Product Record. If an optional PR is included, there might be elements within a composite in the PR that are mandatory. PR.7 for example is optional, but element 7.8 is mandatory if the composite in PR.7 is included.

The ONIX-messages that are exchanged have to be stored by a database solution independently of

whether it consists of one PR or all 25 PR as long as it follows the DTD. This hybrid database solution supports this very well since all information will be stored in the database even if it is not important. When an ONIX-message is received the information will be divided between the three places to store the information. The important information is stored in tables, and since this type of information is usually present in the ONIX-messages the fields within the tables will not be empty. The remaining, not so important, information is stored in the BLOB- and Dictionary table.

To be able to decide which elements that are stored in relational tables, the book industry and the book stores have to look at what kind of information they find important. In section 10.6 there will be a list of information two book stores find important. I will use this list when I illustrate how the information from the ONIX-message in Appendix A can be stored in the hybrid database model.

10.3 The Dictionary table

The Dictionary table looks like this:

record-Reference	Section-Number	section	part	Sequence-Number	Field-Number	field-Name	field-Value	IB	ID	MB	MD

(IB=insertedBy, ID=insertDate, MB=modifiedBy, MD=modifiedDate)

The primary key for the Dictionary table is the three columns recordReference, sequenceNr and fieldNumber. The recordReference is the primary key in the Product table, and identifies the whole product record. In the Dictionary table the recordReference is a foreign key from the Product table, and connects the two tables together. A description of the Product table will come in section 10.8. The sequenceNumber counts the times the fieldNumber for that specific recordReference occurs. The fieldNumber identifies the element within the PR the fieldValue is from, like 10.7 for example. The sectionNr and section are the number and name of the PR where the information is taken from. The part column is the name of the composite, if the information is from a composite. fieldName is the name of the element itself, and in fieldValue the information from the ONIX message is stored. insertedBy and insertDate consist of who and when the information was inserted, and modifiedBy and modifiedDate consist of who and when the record was modified last.

Since the Dictionary table is rather large, and some of the elements (IB, ID, MB, MD) are not very important when illustrating how information is stored, I will not include these columns in the examples below. These columns would basically include the date and person or system who inserted the ONIX-message.

It is important to know that each row in the Dictionary table consists of one element from the PR, and can not consist of a whole PR in the same row as the BLOB table can.

10.4 The BLOB table

The BLOB table looks like this:

record-Reference	section	section-Number	xmlData	insertedBy	insertDate	modifiedBy	modifiedDate

The primary key in the BLOB table are the two columns recordReference and sectionNumber. As mentioned in the description of the Dictionary table is the recordReference a foreign key from the Product table. sectionNumber identifies which PR the information is from, like PR 10, and section is the name of this PR. The columns in the BLOB table are the same as for the Dictionary table, except for the xmlData column. The xmlData column is where the BLOB of XML data is stored. A BLOB could contain the whole ONIX message if we wanted to, but I have designed it to

contain all the elements within a PR do not want stored in the relational tables or in the Dictionary table. In some cases the BLOB can therefore contain a whole PR, while in other cases only a couple of elements within a PR. By having the section columns informing which PR the BLOB is from makes it easier to keep track of which PR that is stored as BLOB. When searching for a whole message or a part of a message stored in a BLOB, it is easier to find it when storing the BLOB after that which the PR information is within. When searching in the BLOB table, you would locate the entire BLOB for a PR by searching for the recordReference that identifies the product and the sectionNumber.

10.5 Information about books

In order to know how to design the hybrid database model, it is important to know what kind of information the book stores and others find important. The publishers need to know what kind of information they have to send to support the receivers' needs. There is no point for the publisher to send tons of information that are not used.

Figure 5.2 in chapter 5 lists information two book store chains think is important to have about a book. In order to place this information in the hybrid database model I have added the PR for each information record the book store chains listed, to place it towards the elements in the ONIX standard. Since the information listed was valued as important by the book store chains, they will have to be stored in the relational tables.

Title (PR 7.8 code 01 in list 15, PR 7.11)	**Original title** (PR 7.8 code 01 in list 15, PR 7.11)
Subtitle (PR 7.8 new code in list 15, 7.11)	**ISBN** (PR 2.7 code 02 in list 5, 2.9)
EAN (PR 2.7 code 03 in list 5, 2.9)	**Subject** (PR 13.9 new code, PR 13.13)
Language (PR 11.3, PR 11.4)	**Published** (PR 20.5)
Edition (PR 10.2, PR 10.3)	**Amount of pages** (PR 12.1)
Price with tax (PR 24.59 code 04 in list 58, PR 24.63)	**Price without tax** (PR 24.59 code 03 in list 58, PR 24.63)
Sales price time (PR 24.74, PR 24.75)	**Minimum order amount** (PR 24.53)
Tax obligated (PR 24.66)	**Picture** (PR 16.7)
Book description (PR 15.3 code 01 in list 33, PR 15.5)	**Audience level** (PR 14.1)
Format of the book (PR 22.1, PR 22.2, PR 22.3)	**Weight of the book** (PR 22.1, PR 22.2, PR 22.3)
Binding (PR 3.1)	**Author** (PR 8.2 code A01 in list 17, PR 8.8, PR 8.10)
Publisher (PR 19.7 code 01 in list 45, PR 19.11)	**Distributor/Supplier** (PR 24.13 code 02 in list 93, PR 24.6
Telephone distributor (PR 24.7)	**Fax distributor** (PR 24.8)
Email distributor (PR 24.9)	**Zip code/Place distributor**
Product owner (PR 19.7 code 06 in list 45, PR 19.11)	**Editor** (PR 8.2 code B01 in list 17, PR 8.8, PR 8.10)
Illustrator (PR 8.2 code A12 in list 17, PR 8.8, PR 8.10)	**Designer** (PR 8.2 code A11 in list 17, PR 8.8, PR 8.10)
Translator (PR 8.2 code B06 in list 17, PR 8.8, PR 8.10)	**Dewey** (PR 13.9 code 01 in list 27, PR 13.13)
Series number (PR 5.7)	**Series** (PR 5.6)
Product group (PR 13.5 new code, PR 13.7, PR 13.8)	**Book group** (PR 13.5 new code, PR 13.7, PR 13.8)
Literature form (PR 13.9 new code, PR 13.13)	**Literature type** (PR 13.9 new code, PR 13.13)
Status message (PR 20.1)	**Statistic group** (PR 13.5 new code, PR 13.7, PR 13.8)

Figure 10.2: The information from figure 5.2 with the PR code from the ONIX standard

Today the ONIX standard does not support all the information listed here. The information is not supported today because the standard needs some additional codes implemented to point towards the Norwegian codes used in the book industry. Information not supported by the standard is the three national code groups that identify the type of book specific to the Norwegian book industry; product group (bokhandlernes varegrouppe in Norwegian), book group (bokgruppe in Norwegian) and statistic group (forlagenes statestikkgrupper in Norwegian).

The literature form and literature type information could is neither supported by the standard today. Whether this should be mapped directly into the standard somewhere or design a national standard for these groups is something that the book industry has to discuss. I have chosen to design new groups for them as I have illustrated in the figure above.

The Zip code / Place for the distributor are also information not supported in the ONIX standard.

You can instead see the city associated with the publisher, or the country and territory where the Distribution Central can distribute the book. This element is however not very important since the ONIX-message primarily contain information about the book, and not location information about the distributor. The logistic system could be used to get hold of this information.

To read what the ONIX standard describe on each of these records, it could be found in the ONIX standard(EDItEUR6 2003). There will below be given a short introduction to each PR to make it easier to understand why I chose to store the information within the PR like I did.

10.6 The tables in the hybrid database model

I will here give a short overview of all the tables that shall be included in the hybrid database solution. To be able to support the information need listed in figure 20, I designed 22 tables including the BLOB and Dictionary table. The following is a list of the tables in the hybrid database solution:

Dictionary table	BLOB table
Header table	Product table
Title table	Contributor table
Contributor role table	EditionType table
Edition table	Language table
Series table	Set table
MainSubject table	AdditionalSubject table
Audience table	SchoolAudience table
Description table	Picture table
Publisher table	Measure table
Supplier table	Price table

Figure 10.3: Tables in the hybrid database solution

The BLOB table, Dictionary table and the 20 relational tables are connected as I illustrated in figure 17. The Product table links all the tables together, and the remaining tables are an independent table connected to the Product table. In the next section I will begin to look at how the information from the ONIX Product Record message in Appendix A are stored in the tables above.

10.7 Storing of an ONIX message

I will in this section look at how the ONIX Product Record message in Appendix A are stored in the tables within the hybrid database model. All of the 25 PR (EDItEUR6 2003) will be explored separately to see how each of the PR are stored in the hybrid database model. The ONIX-message in Appendix A contains only one Product Record, and the information about the product are placed between the <Product> </Product> tags.

A description of how each of the PR is stored in the hybrid database will now be given. I will begin with PR.1, and go through the entire 25 PR from the ONIX for Books [1]

PR. 1 - Record reference number, type and source
PR 1 identifies the record, and informs what type of notification the record gives. The PR is very small, with only 8 non-repeating elements. Two of these elements are mandatory, Record Reference (1.1) and Notification Type (1.2). Record Reference uniquely identifies the Product record, while Notification Type specifies the type of notification or update. These two elements are important in order to identify the ONIX Product Record, while the remaining elements (1.3-1.8) in the PR are not very important. The elements 1.3-1.8 contain information about the record source among other things.

Appendix A contains the following elements from PR.1:

```
<RecordReference>8202231868</RecordReference>
<NotificationType>03</NotificationType>
<RecordSourceType>01</RecordSourceType>
<RecordSourceName>J.W. Cappelens Forlag AS</RecordSourceName>
```

The important elements (1.1-1.2) are stored in a Product table, while the remaining elements (1.3-1.8) not so important are stored in the Dictionary table.

The Product table:

recordReference	notificationType
8202231868	03

The primary key in the Product table is the recordReference column, since this element according to ONIX for books [1] identifies the product record. The ISBN is used as the recordReference in the ONIX-message shown in Appendix A. You do not have to use the ISBN as the recordReference, but this could be another type of unique number. According to an IT Manager from the book industry told me that it is not recommended to use the ISBN as the recordReference, but choose another unique number. This is because you can have the same ISBN multiple times owned or sold by different companies.

The Dictionary table:

Record-Reference	section-Number	section	part	sequence-Number	field-Number	field-Name	field-Value
8202231868	1	Record Reference number, type and source		1	1.5	Record source type code	01
8202231868	1	Record reference number, type and source		1	1.8	Record source name	J.W. Cappelens Forlag AS

PR. 2 – Product numbers

PR 2 is also a rather small with only 12 elements. The purpose with this PR is to identify different numbers related to the product. Important information to collect from this PR is the ISBN and EAN. The ProductIDType (2.7) informs about what type of value the IDValue (2.9) contains. The ProductIDType refers to list 5 in the ONIX Code List (EDItEUR5 2003), where it is written what type of IDValue the ProductIDType refers to. ProductIDType 02 from list 5 in the ONIX Code List informs that the IDValue is an ISBN-10, and 03 is an EAN-13. The IDValue (2.9) informs what the value of the ProductIDType is, in this example what the value of the ISBN and EAN. The ISBN and EAN for a book are unique numbers, and can therefore be stored in the Product table since the book store chains found this information important. The remaining elements in PR.2 (2.1-2.8, 2.10-2.12) and 2.9 when the ProductIDType is neither 02 nor 03 are stored in the Dictionary table.

The following elements in Appendix A are from PR.2:

```
<ProductIdentifier>
  <ProductIDType>02</ProductIDType>
  <IDValue>8202231868</IDValue>
</ProductIdentifier>
<ProductIdentifier>
  <ProductIDType>03</ProductIDType>
  <IDValue>9788202231866</IDValue>
</ProductIdentifier>
```

The Product table we designed in PR.1 will then look like:

recordReference	notificationType	ISBN	EAN
8202231868	03	8202231868	9788202231866

PR 3 Product form
PR 3 is not as little as the previous 2 PR, but contains 31 elements within several composites. This PR contains mainly information about what type of form the product has, and if the product consists of several parts. Product form (3.1) is the only important information from PR.3. Since this element is mandatory and non-repeating it is stored in the Product table designed in PR 1. The remaining elements (3.2-3.31) are stored in the Dictionary table.

The following element in Appendix A is from PR.3:

```
<ProductForm>BB</ProductForm>
```

The Product table designed in example 1, and modified in PR.2 will then look like:

recordReference	notificationType	ISBN	EAN	productForm
8202231868	03	8202231868	9788202231866	BB

PR 4 Epublication form
PR 4 is a small PR, with only 10 elements. It is only used when the product has been identifies as DG in PR 3. Information regarding Epublication was not mentioned by the book store chains, so it is not very important. The whole PR is stored as a BLOB in the BLOB table. The ONIX-message in Appendix A did not contain any information from this PR, so nothing will be added to the BLOB table in this example. In PR 9 I will show how XML data are stored in the BLOB table.

PR 5 Series and PR 6 Set
PR 5 is used if a product belongs to a group of other products, where the group has not a definite number of products or within a given period. PR 6 on the other hand is also a group of products, but it has a definitive amount of products and within a given time. These two PR are not included if the product are not in relation to any series or set.

Information about Series was mentioned by the book store chains as important, so the title (5.6) and the number within the series (5.7) from PR.5 are stored in a Series table, and the title (6.6), set part number (6.7) and set part title (6.8) from PR 6 are stored in a Set table. The remaining elements from the two PR are stored in the Dictionary table. Appendix A did not contain any information from these PR, but there will still be an illustration how the Series table would look like. I will use the product reference 1234567890 so there will not be any confusion with the ONIX-message in Appendix A.

The Series table:

recordReference	seriesTitle	numbersWithinSeries
1234567890	Mountain	Volume 3

PR 7 Title

PR 7 is a normal size PR, with 23 elements. Within this PR information regarding the book title, original title and subtitles could be stored. Some of the information within this PR is important, while others could be stored in the Dictionary table.

This PR consists of three composites with elements; Title (7.8-7.14), Work identifier (7.15-7.17) and Website (7.18-7.20), which are all repeatable, and some optional and non-repeatable elements 7.1-7.7 and 7.21-7.23, which do not belong to any composite. The elements 7.1-7.7 are the same elements that are inside the title composite with the only difference being that they are not repeatable and are deprecated.

To be able to support the information listed above as important, a title table are created. The Title table contains the title type (7.8) and title text (7.11). Some of the elements outside the title composite are deprecated and not recommended to use. However if these elements have some value they are converted into the title composite and the two elements 7.8 and 7.11 stored in the Title table.

To make sure that the remaining elements could be searched for or updated even if they are not stored in the Title table; they are stored in the Dictionary table.

```
<Title>
  <TitleType>01</TitleType>
  <TitleText>Delvis til stede</TitleText>
  <Website>
    <WebsiteDescription>J.W. Cappelens Forlag. Katalog</WebsiteDescription>
    <WebsiteLink>http://www.cappelen.no/main/katalog.asp?isbn=8202231868
    </WebsiteLink>
  </Website>
</Title>
```

The Title table:

recordReference	titleType	titleText
8202231868	01	Devis til stede

The primary key for the Title table would be all the three columns, since the titleType could have the same type of title repeated several times. recordReference is the foreign key as it is for all the other relational tables. The reason for not including the subtitle element (7.14) is because this is a non-repeatable element, and the IT-consultant at FS told me that many books have several subtitles. She thinks that we have to add a new title type in the code list with the name subtitle, so it is possible to have several subtitles.

The Dictionary table:

Record-Reference	section-Number	section	part	sequence-Number	field-Number	field-Name	field-Value
8202231868	7	Title	Website	1	7.19	Website description	J.W. Cappelens Forlag. Katalog
8202231868	7	Title	Website	1	7.20	Link to website	http://www.cappelen.no /main/katalog.asp? isbn=8202231868

PR. 8 - Authorship
PR 8 is a large and complex PR with 34 elements divided between several composites. Information within this PR is the name of the persons who contributed to the product and which part they had. Storing information regarding the authorship of a book is a bit more complicated, since there are some dependencies between which elements that have to be included. The PR consists of the elements 8.1-8.34, where the elements 8.1-8.32 make a big composite, with five smaller ones inside. Interesting information within the PR is the name of the persons who have contributed to the book, and the role he had. The most important information is stored in the Contributor table and ContributorRole table, while the remaining information is stored in the Dictionary table. The ContributorRole table shows the role the Contributors to the book had, and the Contributor table list the name of all the contributors to the book.

The following elements in Appendix A are from PR.8:

```
<Contributor>
  <SequenceNumber>1</SequenceNumber>
  <ContributorRole>A01</ContributorRole>
  <NamesBeforeKey>Ingvar</NamesBeforeKey>
  <KeyNames>Ambjørnsen</KeyNames>
  <Website>
    <WebsiteDescription>J.W. Cappelens Forlag. Forfatterregister</WebsiteDescription>
    <WebsiteLink>http://www.cappelen.no/main/forfatter.asp?f=7000</WebsiteLink>
  </Website>
</Contributor>
```

The elements 8.1, 8.4-8.6, 8.8, 8.10-8.11 are stored in the Contributor table, the ContributorRole Table contain element 8.1 in case several contributors have the same role and element 8.2. The remaining elements (8.3, 8.7, 8.9 and 8.12-8.34) are stored in the Dictionary table.

The Contributor table:

record-Reference	sequence-Number	sequence-Number-WithinRole	person-Name	Person-Name-Inverted	Names-BeforeKey	keyNames	names-AfterKey
8202231868	1				Ingvar	Ambjørnsen	

There are several columns within the Contributor table that are almost the same. This is not necessary if the parties within the book industry agree upon how the information is filled out. The book industry has to decide which fields to use when sending this type of information. The Contributor table above illustrates how some elements are used.

The ContributorRole table:

recordReference	contributorRole	sequenceNumber
8202231868	A01	1

The Dictionary table:

record-Reference	section-Number	section	Part	sequence-Number	field-Number	field-Name	field-Value
8202231868	8	Author-ship	Website	1	8.29	Website-Description	J.W. Cappelens Forlag Forfatter-register
8202231868	8	Author-ship	Website	1	8.30	Website-Link	http://www.cappelen.no/main/forfatter.asp?f=7000

PR. 9 - Conference
PR 9 is a PR of normal size, but since the content depends on that the product has a relation a conference, this is not a very much used PR. Information within this PR is about one or more conferences with a relation to the product. Most products do not have any value in this PR, since there are few books with a relation to a conference. The book store chains did not mention this type of information as important in the list they gave me either, so the whole PR is stored as a BLOB in the BLOB table. However if a company is very interested in products with a relation to conferences, they could design tables where this type of information is stored.

There are no elements in Appendix A that are from PR.9, but I will give an example of what information within this PR looks like and what are stored in the BLOB table:

```
<Conference>
  <ConferenceRole>The role between product and conference</ConferenceRole>
  <ConferenceName>The name<ConferenceName>
  <ConferenceDate>The date<ConferenceDate>
  <ConferencePlace>The place<ConferencePlace>
</Conference>
```

The BLOB table:

record-Reference	section	section-Number	xmlData	inserted-By	Insert-Date	modifiedBy	modifiedDate
8202231868	Conference	9	<Conference> <ConferenceRole> The role between product and conference </ConferenceRole> <ConferenceName>The name<ConferenceName> <ConferenceDate>The date <ConferenceDate> <ConferencePlace>The place<ConferencePlace> </Conference>	RH	01.01.2005		

PR. 10 - Edition
PR 10 is a PR within two different areas. The first five elements (10.1-10.5) are concerning the edition a product has, while the remaining elements (10.6-10.16) are concerning religious text. A product's relation to religious text is not listed as important and will most likely never be searched for can therefore be stored as a BLOB in the BLOB table.

The book store chains mentioned that information regarding the edition is important for them to know. The Edition type code (10.1) is repeatable and is therefore stored in an EditionType table with the foreign key from the Product table, recordReference. The edition number (10.2) and edition version number (10.3) are stored in Edition table, but what about edition statement (10.4) and no edition statement (10.5)? Element 10.4 and 10.5 are stored in either the Edition table or the Dictionary table. The elements do not seem very important, and since the book store chains have not mentioned this as important, they are stored in the Dictionary table. If the book stores change their minds regarding these elements, there is no problem in moving them to the Edition table.

The following elements in Appendix A are from PR.10:

```
<Edition>
  <EditionNumber>1</EditionNumber>
</Edition>
```

According to the ONIX for books (EDItEUR6 2003) the Edition composite does not exist, but are only elements within the Edition PR. The statement from Appendix A is therefore wrong because

it contains an <Edition> composite.

Anyway, Appendix A only contained one element from PR.10, which makes it difficult to illustrate how all the three table types are used I will therefore design how the elements from PR.10 looks like:

```
<EditionTypeCode>REV</EditionTypeCode>
<EditionTypeCode>ILL</EditionTypeCode>
<EditionNumber>1</EditionNumber>
<EditionVersionNumber>2</EditionVersionNumber>
<EditionStatement>Some kind of statement</EditionStatement>
<Religious Text>
  <Bible>
    <BibleContents> OT</BibleContents>
    <BibleVersion>JER</BibleVersion>
    <BibleVersion>AMP</BibleVersion>
  </Bible>
</Religious Text>
```

The information could then be placed in the two relational tables, the Dictionary table and the BLOB table, and would look like the following:

The EditionType table:

recordReference	editionTypeCode
8202231868	REV
8202231868	ILL

The Edition table:

recordReference	editionNumber	editionVersionNumber
8202231868	1	2

The Dictionary table:

record-Reference	section-Number	Section	Part	sequence-Number	field-Number	field-Name	field-Value
8202231868	10	Edition		1	10.4	Edition Statement	Some kind of statement

The BLOB table:

record-Reference	section	section-Number	xmlData	inserted-By	Insert-Date	modifiedBy	modifiedDate
8202231868	Edition	10	<Religious Text><Bible> <BibleContents>OT</BibleContents> <BibleVersion>JER</BibleVersion> <BibleVersion>AMP</BibleVersion> </Bible></Religious Text>	RH	01.01.2005		

PR. 11 - Language

PR. 11 is a small PR with five elements, but still informs which language a product has. The Language PR is a rather important, so the information within the PR is stored in a language table. The elements 11.1 and 11.2 are deprecated, and are the same as can be in the language composite (11.2-11.5). The language table consists of the three elements 11.3-11.5, where 11.1 and 11.2 is combined.

The following elements in Appendix A are from PR.11:

```
<Language>
  <LanguageRole>01</LanguageRole>
  <LanguageCode>nor</LanguageCode>
</Language>
```

The Language table:

recordReference	languageRole	languageCode	countryCode
8202231868	01	Nor	

PR. 12 Extents and other content

PR 12 is a small PR with only 12 elements. The most important information stored in this PR is concerning how many pages the product has, or illustrations. From PR 12 there is only one important element (12.1), which is the number of pages. The other elements (12.2-12.12) are other forms of writing the number of pages and information concerning illustrations. Since this type of information is not mentioned in the information list above, they are stored in the Dictionary table in case someone wants to search or update the information. The information about the number of pages (12.1) a product has is a non-repeating and is therefore stored in the Product table instead of a separate table only for this element.

The following element in Appendix A is from PR.12:

```
<NumberOfPages>182</NumberOfPages>
```

The Product table designed in PR.1, and modified in PR.2 and PR.3 would then look like:

recordReference	notificationType	ISBN	EAN	productForm	numberOfPages
8202231868	03	8202231868	9788202231866	BB	182

PR. 13 Subject

PR 13 is a PR of normal size and contains information about what kind the main subject and additional subject of the product is about according to codes listed. A product's product group identifies the type of category the books fits. Since the book store chains have listed a book's groups as important, two tables are designed from the important element in this PR, the MainSubject table (13.5-13.8) and the AdditionalSubject table (13.9, 13.11-13.13). The MainSubject table contains a description of the main subject classification or subject heading, and the AdditionalSubject table contains a description of a subject classification or subject heading.

Today the Main subject scheme identifier (13.5) and the subject scheme identifier (13.9) in the two tables do not contain the local codes for Norway. This has to be requested by the book industry in Norway and added by EDItEUR. The Subject scheme version (13.6) identifies the version or edition of the scheme specified in the Main subject scheme identifier. The Subject code (13.12) identifies the class or category code from the scheme in the Main subject scheme identifier, and the Subject heading text (13.8 and 13.13) is the name of the code specified in either the main subject scheme identifier or the Subject code.

If the code system for books in Norway needs additional columns to identify the right book, this has to be requested to EDItEUR in order for them to add additional columns.

Appendix A did not contain any elements from this PR, so the two tables below are empty, except for a value I have added in one of the tables. The subject scheme identifier 01 stands for Dewey, which is a classification type set by librarians.

The MainSubject table:

recordReference	mainSubject-SchemeIdentifier	subjectSeriesVersion Number	subjevtCode	subjectHeadingText

The AdditionalSubject table:

recordReference	Subject-SchemeIdentifier	subjectSchemeVersion	subjevtCode	subjectHeadingText
	01			

PR 14 Audience
PR 14 is a small PR that identifies what kind of people the product is for, and for schoolbooks, which grade in school that uses that book. Important information in the Audience PR is who the book is for, and this is covered by element 14.1. Since this element is repeatable, the information is stored in a separate Audience table instead of having it stored in the Product table.

The book store chains mentioned audience as important information, and we talked about the audience of school books as also being important. Information about the audience level concerning school books is covered by the audience composite (14.7-14.11). Codes for the Norwegian school system are not covered in the code list for the audienceRangeQualifier (14.7), but only the US grade system. The Norwegian school system have to be added or a more generalized European grade system be created.

The remaining elements in the PR (14.2-14.6, 14.12-14.14) are stored in the Dictionary table.

Appendix A did not contain any information from this PR, but I will illustrate how the table could look.

The Audience table, where one of the products have two audienceCodes connected:

recordReference	audienceCode
1234567890	01
1234567890	05
2345678901	03

The SchoolAudience table:

recordReference	audienceRange-Qualifier	AudienceRange-Precision	AudienceRange-Value	AudienceRange-Precision	AudienceRange-Value
8202231868	11	03	1	04	3

The range qualifier 11 in the above example refers to the US school range. This school system is a bit different than the Norwegian, so the grades are not similar to the Norwegian grades. However, in the example it says that the product is for the ones from the 1st grade to the 3rd grade.

PR 15 Descriptions and other supporting text
PR 15 is a PR of normal size with 15 elements. The PR is used when different types of descriptions about a product are given. Descriptions about a product are according to the book store chains of interest. A Description table contains what kind of description it is (15.3) and the description itself (15.5), while the remaining information (15.1-15.2, 15.4, and 15.6-15-14) are stored in the Dictionary table.

The following elements in Appendix A are from PR.15:

```
<OtherText>
  <TextTypeCode>02</TextTypeCode>
  <Text textformat="04">En mørk novellesamling om forvirrede,
        løgnaktige og ødelagte menn. Mer handling og dialog
        enn i tidligere samlinger. Dramatiske, intense og sterke,
         også når det gjelder forståelse av identitet.
  </Text>
</OtherText>
```

The Description table:

recordReference	otherTextTypeCode	otherText
8202231868	02	En mørk novellesamling om forvirrede, løgnaktige og ødelagte menn. Mer handling og dialog enn i tidligere samlinger. Dramatiske, intense og sterke, også når det gjelder forståelse av identitet.

The Other text type code 02 indicates that the text is a short description. Other text types can be main description, review text and author comments among many other.

PR 16 Links to image/audio/video files
PR 16 is also a PR of normal size and is used if different kinds of picture are going to be sent. In this PR there will be information about image or audiovisual files. The two book store chains were interested in receiving a picture of the book, which means that a Picture table contains the element 16.4-16.5, 16.7-16.8. The remaining elements within the PR are stored in the Dictionary table.

There is no information of this kind in Appendix A, but I will illustrate how the Picture table looks like.

The Picture table:

Record-Reference	image/audio/video file type code	image/audio/video file format code	image/audio/video file link type	image/audio/video file link
1234567890	04	02	01	www.cappelen.no/pictures/bilde1.gif
1234567890	12	03	06	Bilde2.jpeg

The file type code 04 stands for front cover image, and 12 is the Product logo. The format codes 02 stands for gif, and 03 for JPEG. The file link type is an URL if it is 01 and filename if it is 06.

PR 17 Prizes
PR 17 is a very small PR with only 6 elements. This PR would most likely not be used, since it contains information about prizes and awards related to a product. This kind of information can be stored in a BLOB table or in the Dictionary table, as it is not important for the book stores. I choose to store the information from this PR in the Dictionary table, so the information are more easily be searched for or updated and because this type of information will almost never be inside an ONIX-message.

There were no elements in Appendix A from PR.17.

PR 18 Content items
PR 18 is a PR of normal size and with many composites. The information stored in this PR is descriptions about a product, made in a hierarchy. This type of information is of little or no importance, so it is stored in the BLOB table. There were no elements in Appendix A from PR.18.

PR 19 Publisher
PR 19 has a normal size with 19 elements. The PR contains information about the book's publisher.
Important information from this PR is the role, name, city and country (19.7, 19.11 and 19.15-19.16) a publisher has. This information is stored in a Publisher table, since this is information of interest among the book stores. The remaining elements in PR 19 are stored in the Dictionary table.

The following elements in Appendix A are from PR.19:

```
<Publisher>
  <PublisherName>J.W. Cappelens Forlag AS</PublisherName>
  <CityOfPublication>Oslo</CityOfPublication>
  <CountryOfPublication>NO</CountryOfPublication>
  <Website>
    <WebsiteDescription>J.W. Cappelens Forlag WebSite</WebsiteDescription>
  </Website>
</Publisher>
```

The Publisher table:

recordReference	publishingRoleCode	publisherName	cityOfPublication	countryOfPublication
8202231868		J.W. Cappelens Forlag AS	Oslo	NO

If a book is imported from a foreign country by a publisher, the product owner is the company from which the publisher got the book from. This information is stored in the Publisher table, where the publishing role code 06 stands for published on behalf of. Code 01 stands for publisher.

The Dictionary table:

record-Reference	Section-Number	section	Part	sequence-Number	field-Number	field-Name	field-Value
8202231868	19	Publisher	Website	1	19.13	Website Description	J.W. Cappelens Forlag WebSite
8202231868	19	Publisher	Website	1	19.14	Link to website	http://www. cappelen.no/

PR.20 Publishing status and dates, and copyright
PR 20 has a normal size and contains information regarding when a book was published and the status of the book. Information about the publishing status both code, text that informs the status and the publishing date (20.1, 20.2 and 20.5) are stored in a Publishing table, since this is information similar to the information listed by the book store chains. However, this information is non-repeatable and is therefore stored in the Product table.

The Dictionary table stores the other information within the PR.

The following elements in Appendix A are from PR.20:

```
<PublicationDate>20031009</PublicationDate>
```

The Product table designed in PR.1 and modified in PR.2, PR.3 and PR.12 would then look like:

Record-Reference	NT	ISBN	EAN	Product-Form	Number-OfPages	PS	PSN	Publishing-Date
8202231868	03	8202231868	9788202231866	BB	182			20031009

(NT=notificationType, PS=publishingStatus, PSN=publishingStatusNote)

PR.21 Territorial rights and other sales restrictions

PR 21 is of normal size and contains information concerning which rights there are when it comes to selling the book different geographically places. This type of information is not listed as important by the two book store chains, and is therefore stored in the Dictionary table or the BLOB table if it appears in an ONIX-message. Since it is not likely that many ONIX-messages have any information concerning this type of information, it is stored in the Dictionary table. By storing the information there, it would be easy to search for or update if present.

There were no elements in Appendix A from PR.21.

PR.22 Dimensions

PR 22 is a small PR with important information. To know how the product looks like, and how many mm it has in height, width, thickness and weight is important for the Distribution Centrals, since they distribute the books on behalf of the publishers.

Information about a product's measurements is according to the two book store chains listed as important. A Measure table contains information about what type of measure it is, the measurement and the unit code (22.1-22.3), and the Dictionary table contains the rest.

The following elements in Appendix A are from PR.22:

```
<Measure>
  <MeasureTypeCode>01</MeasureTypeCode>
  <Measurement>205</Measurement>
  <MeasureUnitCode>mm</MeasureUnitCode>
</Measure>
<Measure>
  <MeasureTypeCode>02</MeasureTypeCode>
  <Measurement>130</Measurement>
  <MeasureUnitCode>mm</MeasureUnitCode>
</Measure>
```

The Measure table:

recordReference	measureTypeCode	Measurement	measureUnitCode
8202231868	01	205	mm
8202231868	02	130	mm

The measureTypeCode 01=height, 02=width. It is also possible to store the measures of the thickness and weight among other things.

PR.23 Related products
PR 23 is a large PR with 34 elements and several composites. Information within this PR is about identifying related products of any type. The book store chains have not mentioned this information as important, so the whole PR is stored in the Dictionary table.

There were no elements in Appendix A from PR.23.

PR.24 Supplier, availability and prices
PR 24 contains 77 elements and is therefore very large. This PR contains information concerning the distribution of books, where information about the Distribution Central and the prices are listed. It also contains information like the availability of books and return agreements.

Important information in this PR based upon the list above is stored in two tables; the Supplier table and the Price table. The Supplier table contains the supplier name and role (24.6, 24.13), and the Price table contains different prices the book store chains found important, minimum order, the currency code, tax rate and when the book could be sold for a sales price (24.53, 24.63, 24.64, 24.66, 24.74, 24.75). In element 24.49 there are two price types in list 58 from the Code list [2] an IT-Manager from the book industry told me the book industry is interested in. I will therefore use these two types from list 58 (RPR including sales or value-added tax if applicable, fixed retail price including tax) and create two price fields of them where the price is stored. If the ONIX-message contains a different kind of price type, it is stored in the Dictionary table.

The remaining elements within this PR are stored in the Dictionary table. The reason for not storing the telephone number, fax number and email address in the Supplier table is because these elements are repeatable. By storing them in the Dictionary table, a new table would not have to be designed.

The following elements in Appendix A are from PR.24:

```
<SupplyDetail>
  <SupplierName>SentralDistribusjon ANS</SupplierName>
  <TelephoneNumber>+47 22985700</TelephoneNumber>
  <FaxNumber>+47 22985720</FaxNumber>
  <EmailAddress>sdinfo@sd.no</EmailAddress>
  <SupplierRole>02</SupplierRole>
  <SupplyToCountry>nor</SupplyToCountry>
  <Website>
    <WebsiteDescription>SentralDistribusjon ANS</WebsiteDescription>
    <WebsiteLink>http://www.sd.no/</WebsiteLink>
  </Website>
  <AvailabilityCode>CS</AvailabilityCode>
  <Price>
    <PriceTypeCode>04</PriceTypeCode>
    <PriceAmount>329.00</PriceAmount>
    <CurrencyCode>NOK</CurrencyCode>
  </Price>
</SupplyDetail>
```

The Supplier table:

recordReference	supplierName	supplierRole
8202231868	SentralDistribusjon ANS	02

The Price table:

Record-Reference	rpr-WithTax	retailPrice-WithTax	minimum-Order-Quantity	currency-Code	taxRate	Price-EffectiveFrom	price-EffectiveUntil
8202231868		329		NOK			

The Dictionary table:

Record-Reference	section-Number	section	part	sequence-Number	field-Number	field-Name	field-Value
8202231868	24	Supplier, availability and prices	Supplier and trade data	1	24.7	Supplier telephone number	+47 XXXXXXXX
8202231868	24	Supplier, availability and prices	Supplier and trade data	1	24.8	Supplier fax number	+47 XXXXXXXX
8202231868	24	Supplier, availability and prices	Supplier and trade data	1	24.9	Supplier email address	sdinfo@sd.no
8202231868	24	Supplier, availability and prices	Supplier and trade data	1	24.14	Supply-to country	Nor
8202231868	24	Supplier, availability and prices	Website	1	24.11	Website description	SentralDistribusjon ANS
8202231868	24	Supplier, availability and prices	Website	1	24.12	Link to website	http://www.sd.no/
8202231868	24	Supplier, availability and prices	Supplier and trade data	1	24.21	Availability status code	CS

PR 25 Sales promotion information
PR 25 is a very small PR with 5 non-repeating elements. This PR consists of information regarding sales promotion, which has not been listed as important by the two book store chains. The whole PR is therefore stored in the Dictionary table or the BLOB table, depending on how important the book industry finds the information within the PR.

There were no elements in Appendix A from PR.25.

10.8 Comments to the hybrid database solution

There is no right or wrong answer to how the information should be stored in the hybrid database, but it depends on the information need the companies have. The solution I presented here is based upon a list of important information two book store chains gave me. Most of the information they valued was stored in relational tables, while the remaining information was stored in the BLOB- or Dictionary table. If the information is not valued among the book stores, there is no point in having the publishers send it. But if the publishers send it, it has to be stored in one of these tables.

The hybrid database solution required 22 tables in order to support the information that the book store chains value. This is not a high amount of tables compared to how many tables the solution would require if only relational tables were going to be used, and not a BLOB table and a common Dictionary table. The good news about the solution presented here is that it does not have to be the final solution. If the parties see that they need some additional elements stored in a relational tables, this is easy to do.

Chapter 11
How two relational databases support XML

In the previous two chapters I explored what kind of database solution that best support storing XML based ONIX-messages. I also illustrated how a hybrid database model could be used to store the information from an ONIX-message.

In this chapter I will explore how two relational databases support XML, to see if the ONIX-messages could be stored in one of these solutions as an alternative to the hybrid database in the previous chapter.

The two solutions I will explore are the Oracle database and the Microsoft SQL server.

Both Oracle and Microsoft SQL servers have solutions that give rich support for XML, and are therefore a good alternative for storing of ONIX messages. With the new support of XML it is possible to store the XML document in a relational table, instead of shredding a message into different parts, and then generating the message back together again. In the following sections I will give a description of how XML is supported in both Oracle and SQL server, to see if the ONIX-messages are easier stored using one of them.

11.1 The Microsoft SQL server

The SQL server on the market today is the SQL server 2000. The SQL server 2005 has much better support of XML than the current version, but this is not available yet.

In SQL server 2000 XML is supported by having a simple relation-to-XML mapping:

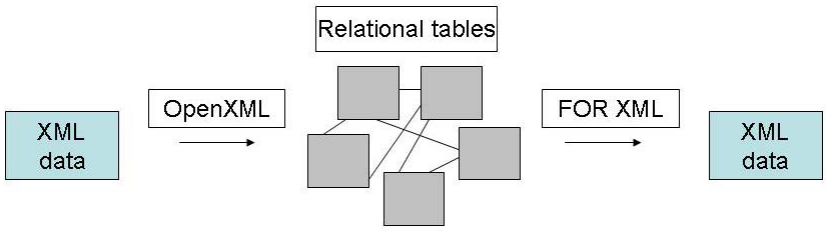

Figure 11.1: XML in SQL 2000

By using a function called OpenXML applications can shred incoming XML data into tables for querying. To generate the XML data from the tables a FOR XML clause in the SELECT statement is used. This is very good for data interchange and web service applications. However, even if there is rich support for mapping between tabular and XML data in the SQL server 2000, there is no support for document order of neither XML documents nor recursive XML scheme (Pal, Fussell et al. 2004).

The features in the new SQL server 2005 will include native implementation for XML storage, indexing and query processing. Existing features in SQL server 2000 like FOR XML and OpenXML will also be enhanced. By introducing a built-in support of XML in forms of the XML data type in the SQL server 2005, the XML support will become much better. You will be able to store native XML data on the SQL server itself, and you can integrate XML with the SQL server database engine. This means that once you have stored the XML data in the SQL server, you will be able to query and update the data, index the XML, and use the query language XQuery to search among the data (Pal, Fussell et al. 2004).

11.1.1 XML data type

XML is a native data type that will be introduced in SQL server 2005. The XML data type is built in the same way as other data types like int and varchar, and can be used like you any other data type, for example in functions, columns and parameters. The data type preserves document order, and can support recursive XML scheme. However, the XML data type is also a user defined data type (UDT), which includes methods for querying and updating XML in variables or columns.

Storing XML data in a relational database can easily be done with the SQL server 2005, since the data type will make it possible to store the XML data as BLOB in columns next to relational columns in the same table. By storing the XML data this way it will enable the XML model to support document order and recursive structures (Pal, Fussell et al. 2004).

The integration of the XML data type with the SQL server's storage and programming models makes it possible to query and update the XML documents. It is also possible to do a join between the XML data and the relational data. The new SQL server 2005 uses the same query engine and optimizer for querying XML data and relational data, which makes it easier to work with XML data. The good querying and data modifications are not available for XML data in the current SQL server 2000.

There are 5 built-in methods for the XML data type, which shall support the querying and modifying of XML instances. It is possible to index the XML columns to make the querying more efficient. When doing a XML indexing on a column, a B+ tree is created with an index of all tags, values and paths. Storage and queries are optimized with the help of an XML scheme. When using the XML scheme the XML columns, variables and parameters are typed according to the XML schemes related (Pal, Fussell et al. 2004).

When it comes to data processing of XML, the new SQL server 2005 provides extensive support. The XML can be stored in an XML data type column, either typed according to a scheme, or left untyped (Pal, Fussell et al. 2004).

11.1.2 Storing of XML

There are three alternative ways to store the XML on the SQL server 2005 (Pal, Parikh et al. 2004):

1. Native storage as XML data type
2. Mapping between XML and relational storage
3. Large object storage

1) Native storage as XML data type:
By using this form of storage the XML data will be stored in an internal representation, which preserves the XML content of the data. The document order, hierarchy containment, element and attribute values are supported. However, an exact copy of an XML document is not made as white space and declarations are not saved.

To store the XML data in an XML data type column is suitable when:

- You want an easy way of storing the XML data on the server, while the document order and document structure is supported
- You want to modify and query the XML data
- You want to index the XML data to increase the processing time when doing a query
- The application needs system catalog views to administer the XML data and schemas.

2) Mapping between XML and relational storage:
In this form of storage, XML document are the XML data decomposed into columns in one or several tables perverting fidelity of the data. The hierarchical structure is taken care of, while the orders of the elements are not.

This form of mapping between the XML schemas and tables in the database, and then create a view of the persistent data is suitable when:

- You want to use an XML programming model by using XML view to access relational data
- You want to query or update the XML data from using Xpath in XML view
- You have an XML scheme for your data
- You want to decompose the XML data into underlying tables by using XML view

3) Large object storage:
An exact copy of an XML document is stored. To have an exact copy of a XML document is necessary if it is a legal document.

A hybrid model is often used to store the XML. The hybrid model is a combination of relational and XML data type columns. This means that some XML data can be stored in relational columns, while the rest or the entire XML can be stored in a XML column. A reason for storing some of the XML data in the relational columns is because you want to promote some of the XML data. By using a hybrid model, the performance may be better, since you would have full control over the index in the relational column and the managing of data storage. The important information will be stored in relational table, while less important information can be stored in one or several XML columns.

The hybrid model described here is very much like the hybrid database model I designed in the previous chapter, where some of the XML is stored in relational tables, and some XML is stored in a BLOB table. The difference is that instead of having a BLOB table, the information can be stored in a XML column. When it comes to the Dictionary table I designed is this not present in this solution, since there are only relational columns and XML columns.

11.2 The Oracle database

The Oracle database on the market today is Oracle 10g, while the previous Oracle database was the Oracle 9i release 2.

When the Oracle 9i first came, it introduced several new and improved XML DB features whereas some of them are support of XML scheme, XML DB repository, XPath and XDBUriType (Kristjánsson 2004).

The following description of Oracle database 10g was given by an article about databases support of XML:

Oracle Database 10g breaks new ground in support for XML technology, offering very rich features for importing, storing, querying, and generating XML data. Providing native, structured XML storage as well as support for unstructured document storage and shredding, Oracle Database 10g allows you to pull XML data from files and merge it with relational data in views.
(McCown 2004)

However, the functionality in Oracle 10g is also available in Oracle 9i (McCown 2004).

In the following small sections a description of how XML is supported in the Oracle databases.

11.2.1 Oracle XML DB

The following description was given of Oracle XML DB:

Oracle XML DB is a feature of the Oracle Database. It provides a high-performance, native XML storage and retrieval technology. It fully absorbs the W3C XML data model into the Oracle Database, and provides new standard access methods for navigating and querying XML. With Oracle XML DB, you get all the advantages of relational database technology plus the advantages of XML.

(Oracle2 2005)

The Oracle XML DB accommodates XML content efficiently, and provides native support for XML. It is more efficient to store the XML in the Oracle XML DB instead of in file systems and separate databases when the volume of XML grows. Oracle XML DB can store and manage both structured, unstructured, pseudo and semi-structured data using a standard data model. Usually organizations use different data models depending on what kind of data they want to save. Oracle XML DB also uses standard SQL and XML.

The XML DB supports both structured XML Object-Relational (O-R) storage and Character Large Object (CLOB) storage. The CLOB storage is document fidelity, while O-R storage maintains DOM fidelity by decomposing XML into underlying O-R structures (Kristjánsson 2004).

The Oracle XML DB makes it possible to store, query, update, transform and process XML. It also provides SQL access to the XML data. It has a close integration with relational data, and multiple storage, indexing and retrieval options. There is also support for several query languages and multiple access paths.

The XMLType tables and views storage in the Oracle XML DB are in charge of the native XML storage and retrieval in the database, integrated with SQL. The XML data can be stored in LOB's, in structured storage (O-R), or a hybrid model which is a combination of the two.

11.2.2 XML Developer Kit

Oracle XML DB is a part of the Oracle XML Deverloper's KIT (XDK). The XDK contains a lot of the XML functionality in the Oracle database. The XDK is integrated in the Oracle9i database, and installed when you install the database (Schrag).

The Oracle implementations of the XML 1.0 specification the XDK contains of (Schrag):

- XML parser, which is used to parse and validate an XML document into memory, manipulates the data, and writes the data back to the XML document.
- XSLT processor, which enable you to transform the XML document into another file format document, or another XML document
- XPath engine, which are used by both the XML parser and the XSLT processor

The Oracle unique implementations the XDK contains of (Schrag):

- XSQL page processor and Java servlet, is part of the XSQL page facility. The XSQL page provides a framework for publishing data in XML easy,
- XML SQL utility makes the loading and retrieving of XML documents from the database easier if you use tables, then if you store the whole XML document as a CLOB.

11.2.3 XMLType data type

The Oracle9i database introduced a new data type in release 1, the XMLType. The new XMLType data type was designed to facilitate native handling of XML data in the database, and has some built-in functions which make it possible to create, extract and index XML data stored in the Oracle9i database. The data type can also represent an XML document as an instance in

SQL (Oracle1).

11.2.4 Storing of XML

There are three ways of storing XML using the Oracle database 10g (Lee 2005):

Unstructured: By using an unstructured storage, the whole XML document will be stored as a Character Large Object (CLOB). This type of storage is suitable when there will be no need of doing updates to an XML document, or when it is required to store and retrieve the whole XML document in their entirety. This type of storage provides the highest throughput when inserting or retrieving XML documents (Lee 2005).

CLOB storage is used for storing document-oriented XML document, which I explored together with data-centric XML document in the chapter 9.

By using unstructured storage, the whole document is stored in a column. When using this type of storage the XML document will be retrieved as a whole, and it is not possible to access the CLOB columns directly (Oracle1).

Structured: This type of storage could be used when there is an XML scheme to structure the XML data against. The content of the document is decomposed into a set of objects, based upon the SQL 1999 standard.

This type of storage provides a number of options for storing collections if there is a complex XML document. One of them is that the set of elements in a collection can be stored as XML text in a CLOB column.

An advantage with this method is that it allows you to store XML data inside the database and preserve the hierarchy of the data (McCown 2004).

Shredding: This type of storage, store the XML data from a XML document in relational tables. This form for storage is used when there is a need for access to the data inside the XML document (Scardina). The hierarchical structure of the XML document is lost by using this method (McCown 2004).

Both unstructured and shredded are ok, but limited. The structured type of storage allows you to leverage the power of both relational data and XML hierarchies (McCown 2004).

Storing the XML document using CLOB column is well suited if you do not want to store the collections of element in a column instead of in many tables, or when you want to reduce the parsing and generation overhead related to the sub tree in the document (Lee 2005).

For storing of the ONIX-messages is the structured type of storage preferred, and not the unstructured, since this store the XML document as a whole. If the ONIX-messages were stored as unstructured would they be slow to query, which is not suitable. The ONIX-messages have to have a scheme connected in order to be stored as structured. From EDItEUR's website a schema can be downloaded (EDItEUR2).

Another reason for choosing the structured storing method is because it can store XML data in the database, and still preserve the order of the data, written under the method above.

11.3 Summary of the two databases

After reading an article that highlight how SQL server 2000 and Oracle Database 10g support XML, I want to highlight the following information from the article (McCown 2004):

1. The 4 leading relational databases (Oracle Database, IBM DB2, Sybase ASE and Microsoft SQL Server) do not only store XML, but also hide the complexity with XML.

2. Microsoft SQL Server 2000 falls behind the other three databases when it comes to support of XML. It is for example only possible to store the xml documents as unstructured objects, or shred the XML data info relational tables. Even if the server does not support structured storage, it is possible to parse and write XML documents, by the server's functions. The article mentions that the new version of the Microsoft SQL Server, the 2005 version, will support the storage of structured xml. The SQL server will not be suitable if you want to keep the structure of the XML data, but are ok if you want to store the XML data in relational tables.

3. The SQL server was fast when importing XML files, but the slowest of the four when creating XML files.

4. Oracle 10g provides very rich support of the three storage types (unstructured, structured and shredding). You could merge xml data from files with relational data in views. The Oracle 10g is a head of the other databases when it comes to the support of XML. They do the best job in hiding the complexity of managing XML data; it offers the best query capabilities, and some extra support for schema evolution and WebDAV repositories. Another advantage with the Oracle Database 10g is that it has good features for storing XML data. You can for example manipulate data using familiar SQL tools.

5. The managing of the XML processes was very easy within the Oracle database. When it came to a speed test where thousands of files were imported and created, and large documents imported, the Oracle 10g was very fast. It was best on imports, and came in second in the file creation test.

McCown compares the SQL server 2000, and not 2005 that are soon coming. Even considering the above, the new version of SQL server provides good support for XML, and should therefore be considered as a candidate when it comes (McCown 2004).

As shown in the list above, Oracle got high scores when tested for import and creation which is important for storing XML based ONIX-message since they are of large of size and searches need to be performed quickly. The database was also considered as the database that XML best supported. Using the Oracle database for storage of the ONIX-message would be a very good solution, as an alternative to the hybrid database solution designed in the previous chapter. Even if I have not illustrated how the information from an ONIX-message can be stored in the Oracle database, have I pointed out that this database is good for storing of the XML based ONIX-messages, since the database has a high support of XML.

Chapter 12
Conclusion

This chapter will summarize the finding in this thesis.

I will begin by looking at the hybrid database model designed in chapter 10, and see if the database solution fits the requirements listed in chapter 5.

At the end of the chapter I will see if the objective defined in chapter 1 was answered in this thesis.

12.1 Exploring the solution

The hybrid database solution I designed in chapter 10 showed how the important information given by the two book store chains and the National Library (NL) could be stored. I used the ONIX Product Record Message from Appendix A to illustrate what the information from an ONIX-message could be stored in the hybrid database model. Comparing the information within the ONIX-message from Appendix A and the information listed as important in figure 4.2 and figure 5.2 you will see that most of the information listed in the figures are included in the ONIX-message. The reason for why the ONIX-message from Appendix A does not include some of the important information from the figure in is that some of the information did not exist about that book, and some of the information especially for Norway is not supported by the ONIX standard yet.

The hybrid database model divide the information that can come in an ONIX-message in 3 categories; important, less important and not important. The information within the important category is information the book industry find important. The information is stored in relational tables, to be able to search for this information fast. The information within the less important category is information not listed by the book store chains and the National Library as important, but the information might be of interest since some books may have this information listed about them. This type of information is therefore stored in a Dictionary table, where information that is of some interest, but not often listed in an ONIX-message. The information not important is stored as a BLOB in a BLOB table. This is because this information is never searched for, and not important for the book industry. The reason for storing the information instead of leaving it is to have it if the ONIX-message for example are going to be forwarded to someone else, and they may be interested in this information.

An advantage of the hybrid database solution designed in chapter 10 is that it does not have to be the final one. The database solution can be adjusted after what type of information companies find important has been determined. If for example a company is very interested in books with relation to a conference, they would rather have one or more relational tables about a conference relation to a book, than having this stored as a BLOB in the BLOB table as I did in chapter 10. The database solution can easily be modified to meet different companies' requirements, so the database solution is very flexible.

As mentioned in chapter 10 the ONIX-standard support almost all the information listed in figure 5.2. The only information not supported is the Distribution Centrals' zip code / place. However, this type of information could be collected from the logistic system and not from an ONIX-message where information about products is. Besides this, the standard has to add some Norwegian specific codes to be able to set the right grade in school a book is for, and to be able to send information about what type of book it is using the Norwegian number system for books.

EDItEUR have to add this to the code list if Norway is going to use the standard.

The person interviewed at FS told me that extra code would need to be added in the code list to make it possible to have several subtitles. This was something the companies or the book industry could do themselves.

Even if the standard has to have some extra codes added to fit with Norwegian codes, this is only something that has to be added by EDItEUR. The standard itself supports the information need from the requirements, and probably any additional information other companies in the book industry might have.

12.2 Exploring the requirements

What kind of information two book store chains and the NL found important, was listed in chapter 5. As concluded in the previous section, the international ONIX standard did support almost all this information, but in order to function in Norway, the standard has to add some Norwegian specific codes. Nothing has to be done with the standard itself, but the values the standard use have to be increased to point towards values used in the Norwegian book industry.

Whether the book information listed in figure 5.2 is a complete list over important information or not is difficult to say. However, the ONIX standard explored is an international standard adopted by many companies all over the world. Since the standard supports the requirements these companies have, and it support the requirements the two book store chains listed, it most likely also supports additional information other companies may have.

In chapter 6 were the general requirements regarding a new standard listed in chapter 5 explored. The general requirements were that a standard have to be both flexible and open in order to support the needs in the book industry. From chapter 6 I determined that the ONIX standard was flexible and open, which makes it possible for all the companies in the book industry to adopt the standard, and meet new requirements in the future.

The ONIX standard supports the requirements the book industry has, so they do not have to wonder if there exist a standard that they can use. Based upon the information from the interviews are a standard preferable, so the book industry have to discuss how they are going to implement the standard. As mentioned earlier is there no point in having only a couple of companies implementing the standard, since the advantages from having a standard will become visible when many companies use it.

12.3 Conclusion

In this section I will look at how the thesis has answered the objective determined in chapter 1.

The objective was to determine what kind of information about books needs to be exchanged between publisher and book store, and how this is could be supported by IT.

I interviewed companies from different parts of the book industry, and received valuable information on how the information exchange was working today, and problem areas with how this was done.

Two book store chains and the National Library also gave me a list over book information they found important, and needed to receive and store in their systems.

The interviews uncovered 3 problem areas in the book industry today. One of these was concerning how the information was sent from the publisher to the book store because there was no standard format. This problem area was the only one explored in this thesis. The other two problem areas were concerning what type of information and functions that should be in Mentor and whether a neutral company should be in control of a database with book information about books published in Norway instead of FS. I chose to focus on how the information exchange between the publisher and book store could be improved by using a standard, instead of exploring which information and function to include or not in Mentor. Regarding who the owner of a database with book information should be have I made a suggestion based upon that several of the interviewees wanted a neutral owner, but I have not explored this problem area any further.

A standard makes it easier and more efficient for the publisher to send information to their Distribution Central, for the Distribution Centrals to exchange information, and for the Distribution Central that send information about books to the book store.

I explored the ONIX standard, which was a natural choice since this is an international standard used in several countries. Besides looking a how the standard function, I discussed which database that could be used in order to store information sent using this standard. A hybrid database supported the ONIX standard best. I therefore illustrated how a hybrid database could be used to store information from an ONIX-message, where the information from the book store chains was stored in relational tables. A description of how two concrete relational databases supported XML, and therefore could be used to store the ONIX-message was also given. This was done to show that there are two relational databases that can be used to store the ONIX-messages. After exploring the two relational databases, I saw that the Oracle database supported XML in a better way then the current version of the Microsoft SQL server.

The book store chains informed about information that is important for them to receive and the interviewees told me about their needs regarding how to improve the information exchange between the publisher and book store. By implementing a standard that support the book stores information need, the book industry will easier be able to exchange information with each other. A suggestion of how the information from the ONIX-message could be stored by the receivers were also given, and illustrated.

Appendix A
Example of an ONIX Product Information Message

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE ONIXMessage SYSTEM "http://www.editeur.org/onix/2.1/reference/onix-international.dtd">
<ONIXMessage>

	<Header>
		<FromCompany>Cappelen.no</FromCompany>
		<SentDate>20031215</SentDate>
	</Header>

	<Product>
		<RecordReference>8202231868</RecordReference>
		<NotificationType>03</NotificationType>
		<RecordSourceType>01</RecordSourceType>
		<RecordSourceName>J.W. Cappelens Forlag AS</RecordSourceName>

		<ProductIdentifier>
			<ProductIDType>02</ProductIDType>
			<IDValue>8202231868</IDValue>
		</ProductIdentifier>

		<ProductIdentifier>
			<ProductIDType>03</ProductIDType>
			<IDValue>9788202231866</IDValue>
		</ProductIdentifier>

		<ProductForm>BB</ProductForm>

		<Title>
			<TitleType>01</TitleType>
			<TitleText>Delvis til stede</TitleText>
			<Website>
				<WebsiteDescription>J.W. Cappelens Forlag. Katalog</WebsiteDescription>
			<WebsiteLink>http://www.cappelen.no/main/katalog.asp?isbn=8202231868</WebsiteLink>
			</Website>
		</Title>

		<Contributor>
			<SequenceNumber>1</SequenceNumber>
			<ContributorRole>A01</ContributorRole>
			<NamesBeforeKey>Ingvar</NamesBeforeKey>
			<KeyNames>Ambjørnsen</KeyNames>
			<Website>
				<WebsiteDescription>J.W. Cappelens Forlag. Forfatterregister</WebsiteDescription>
				<WebsiteLink>http://www.cappelen.no/main/forfatter.asp?f=7000</WebsiteLink>
			</Website>
		</Contributor>

		<Edition>
			<EditionNumber>1</EditionNumber>
		</Edition>

		<Language>
			<LanguageRole>01</LanguageRole>
			<LanguageCode>nor</LanguageCode>
		</Language>

		<NumberOfPages>182</NumberOfPages>

		<OtherText>
			<TextTypeCode>02</TextTypeCode>
			<Text textformat="04">En mørk novellesamling om forvirrede, løgnaktige og ødelagte menn.
			Mer handling og dialog enn i tidligere samlinger. Dramatiske, intense og sterke, også
```

```
                    når det gjelder forståelse av identitet.
                    </Text>
                </OtherText>

                <Publisher>
                    <PublisherName>J.W. Cappelens Forlag AS</PublisherName>
                    <CityOfPublication>Oslo</CityOfPublication>
                    <CountryOfPublication>NO</CountryOfPublication>
                    <Website>
                        <WebsiteDescription>J.W. Cappelens Forlag WebSite</WebsiteDescription>
                        <WebsiteLink>http://www.cappelen.no/</WebsiteLink>
                    </Website>
                </Publisher>

                <PublicationDate>20031009</PublicationDate>

                <Measure>
                    <MeasureTypeCode>01</MeasureTypeCode>
                    <Measurement>205</Measurement>
                    <MeasureUnitCode>mm</MeasureUnitCode>
                </Measure>

                <Measure>
                    <MeasureTypeCode>02</MeasureTypeCode>
                    <Measurement>130</Measurement>
                    <MeasureUnitCode>mm</MeasureUnitCode>
                </Measure>

                <SupplyDetail>
                    <SupplierName>SentralDistribusjon ANS</SupplierName>
                    <TelephoneNumber>+47 22985700</TelephoneNumber>
                    <FaxNumber>+47 22985720</FaxNumber>
                    <EmailAddress>sdinfo@sd.no</EmailAddress>
                    <SupplierRole>02</SupplierRole>
                    <SupplyToCountry>nor</SupplyToCountry>
                    <Website>
                        <WebsiteDescription>SentralDistribusjon ANS</WebsiteDescription>
                        <WebsiteLink>http://www.sd.no/</WebsiteLink>
                    </Website>
                    <AvailabilityCode>CS</AvailabilityCode>
                    <Price>
                        <PriceTypeCode>04</PriceTypeCode>
                        <PriceAmount>329.00</PriceAmount>
                        <CurrencyCode>NOK</CurrencyCode>
                    </Price>
                </SupplyDetail>

        </Product>
</ONIXMessage>
```

Appendix B

Oracle and SQL server review

XML Essentials

Oracle leads the pack with an impressive set of XML handling capabilities. IBM provides the essentials, Sybase lacks a few key pieces, and Microsoft just isn't ready for prime time.

	Supports XML shredding	Stores structured XML	Validates schema	Interacts natively with stylesheets	Merges XML with relational data	Supports XQuery	Supports SQL/XML	Supports Web services	Reads/writes to WebDAV repositories
IBM DB2 Universal Database 8.1	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No
Microsoft SQL Server 2000	Yes	No	No	Yes	No	No	Yes	Yes	No
Oracle Database 10g Release 1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Sybase ASE 12.5.1	Yes	No	Yes	No	Yes	No	Yes	Yes	No

SCORECARD

Which Relational Databases Make the XML Grade?

Oracle has the richest, smoothest XML handling capabilities; IBM and Sybase are solid performers; and Microsoft has some catching up to do.

	Consume	Store	Search	Generate
IBM DB2 Universal Database 8.1	Importing XML files into the database was intuitive and fast	Stored XML using all three methods, validated schemas, and fully indexed the data	Easily created views combining XML and relational data but lacked XQuery support	Generated XML files faster than the rest, complete with XSLT transformations
Overall grade B	Grade A	Grade A	Grade B	Grade A
Microsoft SQL Server 2000	Offered many options for pulling out specific data	Inability to store structured XML is a serious deficiency; XML data could be shredded into the database but not validated	Imported XML data is converted to relational data, so granular XML search capabilities are lost	Provided a full set of features to create XML files, but working with the .Net Framework's XML provider is often easier
Overall grade C	Grade A	Grade C	Grade C	Grade B
Oracle Database 10g Release 1	Provided rich methods for consuming XML, including files delivered via HTTP and FTP	Supported all three methods of XML storage plus interaction with WebDAV repositories	Provided the easiest, most comprehensive searching capabilities	Generated XML files quickly and easily; breezed through complex transformations
Overall grade A	Grade A	Grade A	Grade A	Grade A
Sybase ASE 12.5.1	Outstanding job of consuming XML documents	Supported all three flavors of XML storage and provided rich indexing capabilities	Offered many options for creating views and encapsulating queries but no XQuery support	Generated XML files very fast but lacked on-the-fly XSLT transformations
Overall grade B	Grade A	Grade A	Grade B	Grade C

Reference

. Retrieved 30.02, 2004, from http://publishers.asn.au/index.cfm?doc_id=215.

Abiblion. "ONIX Implementation Tutorial - Chapter 1 - Simple XML and ONIX examples." Retrieved 09.02, 2005, from http://abiblion.com/onixtutorial/01.shtml.

Amazon. "ONIX." Retrieved 02.03, 2005, from http://www.amazon.com/exec/obidos/tg/catalog-guide/guide/-/407039/104-2495523-6048757.

APA1. (2003). "APA ONIX Seminar." Retrieved 02.03, 2005, from www.publishers.asn.au/emplibrary/ONIX_Seminar_Nov_2003.ppt.

APA2. "ONIX FAQ." Retrieved 30.08, 2004, from http://publishers.asn.au/index.cfm?doc_id=215.

BISG1. "FAQ - ONIX International." Retrieved 25.01, 2004, from http://www.bisg.org/onix/onix_faq.html.

Bourret, R. (1999-2004). "XML and Databases." Retrieved 05.04, 2005, from http://www.rpbourret.com/xml/XMLAndDatabases.htm.

BSI. (2005). "Types of Standard." Retrieved 23.04, 2005, from http://www.bsi-global.com/HigherEducation/AboutStandards/typesofstandard.xalter.

Champion, M. (2001). "Storing XML in Databases." Retrieved 01.03, 2005, from http://www.eaijournal.com/PDF/StoringXMLChampion.pdf.

Chaudhri, A. B. "Mirror, Mirror on the Wall, What is the Best <XML> Database Solution of All?" Retrieved 11.04, 2005, from www10.org/dday/chaudhri.ppt.

CINCOM. "TOTAL® XML, Choosing an XML Database Solution." Retrieved 11.04, 2005, from http://tiger.cincom.com/PDFs/TOTALXML_Choosing_XML_Database.pdf.

Congress, L. o. "Frequently Asked Questions (FAQ)." Retrieved 01.03, 2005, from http://www.loc.gov/marc/faq.html.

CoverPages2. (2004). "Extensible Markup Language (XML)." Retrieved 09.02, 2005, from http://xml.coverpages.org/xml.html.

CoverPages. (2000). "ONIX International XML DTD." from http://xml.coverpages.org/onix.html.

Dawson, L. (2003). "ONIX in the Library or, Why This Presentation Should Be Longer." Retrieved 01.03, 2005, from www.niso.org/presentations/LauraD.ppt.

DNF. (2004). Retrieved 04.04, 2005, from http://www.forleggerforeningen.no/bokmarkedet_i_norge.htm.

EDItEUR1. "ONIX for Books." Retrieved 20.04, 2005, from http://www.editeur.org/onixfiles2.1/ONIX_Books_DTDs_2.1_rev.02.zip http://www.editeur.org/onixfiles2.1/ONIX_Books_Documentation_2.1_rev.02.zip.

EDItEUR2. Retrieved 02.03, 2005, from http://www.editeur.org/onix.html.

EDItEUR3. Retrieved 02.03, 2005, from http://www.editeur.org/what.html.

EDItEUR4. Retrieved 02.03, 2005, from http://www.editeur.org/ONIX%20International%20FAQ.html.

EDItEUR5 (2003). ONIX Code Lists Issue 2, Recevied from my teacher supervisor, but can also be found in the ONIX standard from the EDItEUR1 reference.

EDItEUR6 (2003). ONIX for Books, Product Information Message, Product Record Format, Release 2,1, rev.01. The new version is within the ONIX standard from the EDItEUR1 reference.

EDItEUR7 (2004). onix-international.dtd, The onix-international.dtd from rev.02 from the EDItEUR1 reference.

EDItEUR8. "ONIX for ISBN." Retrieved 19.02, 2005, from http://www.editeur.org/onix_isbn.html.

EDItEUR9. (2004). "ONIX for Books, ISBN registration format." Retrieved 08.04, 2005, from http://www.editeur.org/040913%20ISBN%20registration%20message.pdf.

EDItEUR10. (2004). "Minutes of the EDItEUR steering group meeting." from http://www.editeur.org/editeur%20meeting%20oct%2004.doc.

EDItEUR11 (2004). Minutes of the EDItEUR steering group meeting held on Monday 15 March 2004 at the London Book Fair, Given by my teacher supervisor.

EDItEUR12 ONIX for Books, Product Information Message, Overview and Data Elements, A document within Rev.02 of the ONIX standard, see reference EDItEUR1.

EDItEUR13 ONIX for Books, Product Information Message, XML Message Specification, A document within Rev.02 of the ONIX standard, see reference EDItEUR1.

Editeurnews (2005). EDItEUR e-NEWSLETTER, February 2005, Received a copy from my teacher supervisor.

Elmasri, R. and S. B. Navathe (2003). Fundamentals of Database systems, Pearson Education.

Flynn, P. (2004). "The XML FAQ." Retrieved 30.08, 2004, from http://www.ucc.ie/xml/.

Forlagsentralen1. "FS Informasjonstjenester." Retrieved 13.02, 2005, from http://www.forlagsentralen.no/1.3.1.htm.

Forlagsentralen2. "FS Distribusjon." Retrieved 13.02, 2005, from http://www.forlagsentralen.no/1.2.1.htm.

Garcia-Molina, H., J. D. Ullman, et al. (2002). Database Systems – The Complete Book, Prentice Hall.

Hanseth, O. "From systems and tools to networks and infrastructures - from design to cultivation. Towards a theory of ICT solutions and its design methodology implications." from http://heim.ifi.uio.no/~oleha/Publications/ib_ISR_3rd_resubm2.html.

Hanseth, O., E. Monteiro, et al. (1996). "Developing information infrastructure: the tension between standardisation and flexibility." Retrieved 08.04, 2005, from http://heim.ifi.uio.no/~oleha/Publications/sthv.html.

HWG.org. (1998). "XML Frequently Asked Questions." Retrieved 09.02, 2004, from http://www.hwg.org/resources/faqs/xmlFAQ.html.

INF5250. (2004). "Information systems in organizations." Retrieved 08.04, 2005, from http://www.uio.no/studier/emner/matnat/ifi/INF5250/h04/.

Intro Graphical view of the ONIX DTD Release 2.1, from the EDItEUR1 reference under the reference folder.

ISBN (2004). Guidelines for the Implementation of 13-Digit ISBNs. Berlin, International ISBN Agency.

Kristjánsson, M. (2004). "Building with Oracle XML Database." Retrieved 12.04, 2005, from http://www.oracle.com/technology/oramag/oracle/04-sep/o54xml.html.

Lee, G. (2005). "Mastering XML DB Storage in Oracle Database 10g release 2." Retrieved 12.04, 2005, from http://www.oracle.com/technology/tech/xml/xmldb/Current/TWP_Mastering_XMLDB_Storage.pdf.

Lotz, W. "ONIX & BISAC Subject Codes." Retrieved 10.04, 2005, from www.bisg.org/docs/BEA04/Wendell_Lotz_BEA_BISG.ppt.

MARC1. "MARC Standards." Retrieved 01.03, 2005, from http://www.loc.gov/marc/.

MARC2. Retrieved 01.03, 2005, from http://www.loc.gov/marc/umb/.

Martin, D. (2004). "Standards for electronic trading and product metadata." Retrieved 02.03, 2005, from www.itale.it/testi/pps/martin.pps.

McCown, S. (2004). "Databases flex their XML." Retrieved 12.04, 2005, from http://www.infoworld.com/article/04/04/23/17FExml_1.html.

Meier, V. (2003). "Envision Tomorrow's Catalog: The Evolution of Cataloging In the E-Commerce Era." Retrieved 30.08, 2004, from http://www.nocall.org/nocallnews/0311envis.html.

Mertz, D. (2001). "Putting XML in context with hierarchical, relational, and object-oriented models." Retrieved 10.01, 2004, from http://www-106.ibm.com/developerworks/library/x-matters8/.

Myers, M. D. (1997). "Qualitative Research in Information Systems." Retrieved 04.04, 2005, from http://www.qual.auckland.ac.nz/.

Nasjonalbiblioteket (2003). FORSLAG TIL ETABLERING AV METADATABASE: CIP-REGISTRERINGER FOR NORSKE UTGIVELSER.

NL. "History, foundation and function." Retrieved 23.04, 2005, from http://libraries.theeuropeanlibrary.org/Norway/page_en.xml.

Oracle1. "Using XMLType." Retrieved 12.04, 2005, from http://www.cs.utah.edu/classes/cs5530/oracle/doc/B10501_01/appdev.920/a96620/xdb04cre.htm.

Oracle2. (2005). Retrieved 11.04, 2005, from http://www.oracle.com/technology/tech/xml/xmldb/index.html.

Pal, S., M. Fussell, et al. (2004). "XML Support in Microsoft SQL Server 2005." Retrieved 11.10, 2004, from http://msdn.microsoft.com/xml/default.aspx?pull=/library/en-us/dnsql90/html/sql2k5xml.asp.

Pal, S., V. Parikh, et al. (2004). "XML Best Practices for Microsoft SQL Server 2005." Retrieved 11.10, 2004, from http://msdn.microsoft.com/xml/buildingxml/xmlanddatabase/default.aspx?pull=/library/en-us/dnsql90/html/sql25xmlbp.asp.

Pascall, M. "XML for Dummies." Retrieved 09.02, 2005, from www.nzetc.org/downloads/xml-for-dummies.ppt.

PublishingTechnologies. (2000). "S3 Support for ONIX." Retrieved 01.03, 2005, from http://www.s3pubtech.com/news/news_092200_ONIX.html.

PublishingTrends. (2002). "The ONIX Odyssey." Retrieved 01.03, 2005, from http://www.publishingtrends.com/copy/0210/0210onix.html.

Scardina, M. "Storing XML Data." Retrieved 16.04, 2005, from http://www.dbazine.com/OLC/scardina1.shtml.

Schrag, R. "XML and Oracle." from http://www.dbspecialists.com/presentations/xml_and_oracle.html.

SD. Retrieved 06.04, 2005, from http://www.sd.no/olfinfo/index.html.

SentralDistribusjon (2005). SD Presentasjon given by the Marketing Director at SD.

Solberg, P. O. (2004). "Sentraldistribusjon kaprer kunder." Retrieved 13.02, 2005, from http://www.bokogsamfunn.no/Nettutgaven/107716/107779.

Suleman, H. (2004). "Information Management, XML and Databases." Retrieved 11.04, 2005, from http://www.cs.uct.ac.za/courses/CS300W/IM/uct_2004_csc303_im_notes_2.ppt.

SwedenStandard exportflt artikel, Given by my teacher supervisor. Explore the elements within the Swedish standard for the book industry.

Valenzuela, D. and P. Shrivastava. "Interview as a Method for Qualitative Research." Retrieved 04.04, 2005, from http://www.public.asu.edu/~kroel/www500/Interview%20Fri.pdf.

W3Schools1. "How can XML be Used?" Retrieved 09.02, 2005, from

http://www.w3schools.com/xml/xml_usedfor.asp.

w3Schools2. "Introduction to DTD." Retrieved 09.02, 2005, from http://www.w3schools.com/dtd/dtd_intro.asp.

W3Schools3. "Introduction to XML Schema." Retrieved 09.02, 2005, from http://www.w3schools.com/schema/schema_intro.asp.

W3Schools4. "Displaying XML with CSS." Retrieved 0902, 2005, from http://www.w3schools.com/xml/xml_display.asp.

W3Schools5. "Displaying XML with XSL." Retrieved 09.02, 2005, from http://www.w3schools.com/xml/xml_xsl.asp.

Williams, K. (2001). "XML for Data: Native XML databases: a bad idea for data?" Retrieved 10.01, 2004, from http://www-106.ibm.com/developerworks/xml/library/x-xdnat.html.