NTNU
Innovation and Creativity

Digital Archive for Scores and Music

Lars Erik Løvhaug

Master of Science in Communication Technology
Submission date: June 2006
Supervisor: Leif Arne Rønningen, ITEM
Co-supervisor: Harald Bjørkøy, Griegakademiet, UiB

Norwegian University of Science and Technology
Department of Telematics

Problem Description

The purpose of the system is to permit publishing of unknown, or unrecorded music (audio, sheet music, etc., possibly video) in a way that makes it available to todays performers and researchers. The system shall have a central server with a access control system and user groups. It shall contain a user friendly library. The access control may also include payment for services. It is possible to include recordings and previously published compact discs in the library.
Tasks:
* The work consists of system integration
* Clarify the needs. Clarify what shall be distributed and who shall distribute it. (Interview with central persons)
* Evaluate existing systems for possible reuse
* Find technical solutions to rights management issues concerning the downloading and use of audio and sheet music from the archive. Suggest ways to protect intellectual property rights and to support payment
* Evaluate digitizing of sheet music using OCR-software
* Specification (or extend existing specification)
* Design (or extend existing design)
* Assess how the implementation can be done

Assignment given: 23. January 2006
Supervisor: Leif Arne Rønningen, ITEM

Music is the wine that fills the cup of silence.

– *Robert Fripp*

Preface

This thesis is the result of the final project of the MSc studies at the Norwegian University of Science and Technology, Faculty of Information Technology, Mathematics and Electrical Engineering, Departments of Telematics.

The study has been carried out at NTNU, Department of Telematics, in the spring semester of 2006.

I would like to thank my mentor at NTNU, Leif Arne Rønningen, for help and guidance during the work with this thesis. The weekly meetings have been very useful for establishing which areas to focus on. I would also like to thank my external mentor at Griegakademiet, University of Bergen, Harald Bjørkøy, for sharing his visions for the system and for proposing the subject for the thesis. A special thanks also goes to Liv Mari Schei for sharing her thoughts in the interviews and to Eirik Refsdal for helpful feedback.

Trondheim, 19th of June 2006.

Lars Erik Løvhaug

Abstract

This thesis establish the requirements for a searchable, on-line archive for music in the form of audio, video and sheet music, and looks at possible ways to design such a system.

The thesis describes and compare existing systems to assess possibilities for reuse. Special attention is given to issues regarding the digital representation of sheet music.

From interviews and evaluation of existing systems requirements for a digital music archive are derived. Furthermore, possible architectures are described. An access management model is developed that support many different use areas. These areas include a public database of music open for public contribution and use, a web-shop function where items in the archive can be charged for, publishing of music and science reports with audio samples, and small-scale collaboration on producing collective documents. All these applications plus other should be possible using the model.

Institutions, organizations or private persons may use this thesis as a design for their digital music archive.

Table of Contents

FIGURES

Tables

Terms and Abbreviations

AAC	Advanced Audio Coding
DRM	Digital Rights Management
GPL	GNU General Public License
IASA	International Association of Sound and Audiovisual Archives
METS	Metadata Encoding and Transmission Standard
MP3	MPEG-1 Audio Layer 3
OCR	Optical character recognition
OMR	Optical Music Recognition
PCM	Pulse Code Modulation
SVG	Scalable Vector Graphics
VBR	Variable Bit Rate
VFR	Variable Frame Rate
XSLT	Extensible Stylesheet Language Transformations
MIDI	Musical Instrument Digital Interface

Part I Background

1 INTRODUCTION

1.1 Background

The background for this thesis is the ongoing initiative of ENOMA (European Network of Online Musical Archives) for making a common network of musical archives for universities and musical institutions in Europe. The thesis provides a background for the development of musical archives suited for such institutions.

Today there are many musical archives spread around the world. The archives range from small personal CD collections to big public collections with many thousand hours of music, video and big sheet music collections. Some of these collections are available on-line, but the majority are collections where the material is stored on physical objects like tapes and disks.

This restricts the possible use of the material. To use it the physical object must be retrieved and then either a user copy is made or the user may borrow it for a period of time. A digital archive system could support the users more effectively because digital information can be copied easier, faster and often more precise than copying analogue material. An on-line digital archive accessible via a computer network can also support users regardless of the users physical placement. The users no longer have to travel to the archive or order copies that are sent in the regular mail. They only need an on-line connection to the archive and may be situated all over the world.

A large amount of sheet music exists but is unavailable to most people. Examples are works of artists that have been dead for a certain number of years. These works are then often in the public domain. The problem lies in the unavailability of the music. Sometimes the prints of the music only exists in a few copies. The music may be free but is hard to find. Making such music available in an web-based system would open up the use of this music. The ability of letting the users of an archive or digital library contribute to the collection is also a possibility in the system outlined in the thesis.

The thesis is meant to be a guideline for the design of an digital library system for music. It is intended to handle the goals of the ENOMA initiative, but where possible, also tries to be general enough to embrace other possibilities.

Preliminary results from this work were presented at an ENOMA Workshop in Bergen in May 2006.

1.2 Motive

The personal motivation for this thesis work is the opportunity of combining my interests in music and computer technology. I have a background in traditional Norwegian music, and have during the last years become more and more interested in the cultural heritage in this music. This has given me knowledge of the many music archives and some of the problems they struggle with.

Digitizing the contents of the archives will make many of the archive tasks easier to accomplish. It will among other things be easier to serve the public and give access to the music. User request for hearing particular recordings can for instance be handled much faster and with less manual work when the collection is stored digitally, properly indexed and accessible through a computer network.

The prospect of making this thesis a useful background document both for the ENOMA project and the traditional music archives of Norway has been the main driving force in my work.

1.3 Problem Statement

The problem statement was originally given in Norwegian, and is included in appendix B in Norwegian.

Digital Archive for Scores and Music

The purpose of the system is to permit publishing of unknown, or unrecorded music (audio, sheet music, etc., possibly video) in a way that makes it available to todays performers and researchers. The system shall have a central server with a access control system and user groups. It shall contain a user friendly library. The access control may also include payment for services. It is possible to include recordings and previously published compact discs in the library.

Tasks:

- The work consists of system integration
- Clarify the needs. Clarify what shall be distributed and who shall distribute it. (Interview with central persons)
- Evaluate existing systems for possible reuse
- Find technical solutions to rights management issues concerning the downloading and use of audio and sheet music from the archive. Suggest ways to protect intellectual property rights and to support payment
- Evaluate digitizing of sheet music using OCR-software
- Specification (or extend existing specification)
- Design (or extend existing design)
- Assess how the implementation can be done

1.4 Methods

To assess the needs for the system a combination of literature studies and qualitative interviews have been used. Three persons and potential users of the system are interviewed. Two of them are also involved in the ENOMA project and have therefore an idea of what the system shall and shall not do.

The author attended an archive conference at Dragvoll, Trondheim in March 2006. The conference gave the author an idea of the status of Norwegian traditional music archives. It also gave insight into the preservation issues at hand.

The descriptions of the existing systems described in Part II are based on literature studies and some testing. The testing includes a small scale test of scanning software for sheet music, tryouts of on-line music archives and tryouts of search systems for music.

The system description in Part III is based on results from part II and constitutes the innovative part of the thesis work.

1.5 Limitations / scope

The ENOMA project was in its startup phase when the work with the thesis started. Therefore some assumptions about how the system in the thesis shall behave and what needs it shall fulfill may be different from the final ENOMA requirements.

The archive system is not designed in detail. The design is outlined, and some central issues are handled in more detail such as user access management and the ownership details of documents and collections.

The database for the documents and metadata is not developed for the system. Before the database can be designed it must be investigated in detail what metadata is needed for the system. This depends on what the system shall be used for and must be specified for each implementation.

The nonfunctional requirements for the system is not fully elaborated in this thesis. This is a task for further studies.

1.6 Organization of the Thesis

The thesis is divided in five main parts.

Part I Background

After an introduction we establish the background for why a digital archive for scores and music is needed. We look at the need to publish material from the viewpoint of two musical genres and also establish needs through interviews.

Part II Existing Systems

This is a background study where we review what already exists. We first look at representation formats for audio, video and sheet music. The emphasis is on possibilities and requirements for a notation language for sheet music. Then we look at how sheet music can be ingested into the archive. Existing archives are then studied for reuse of ideas and technology. Existing software and payment systems are then examined for reuse. Rights management issues are discussed next, followed by a study of metadata issues and different search methods for a digital music archive.

Part III The System

This is where the system is specified. We start with summing up the needs from the background study and discuss the requirements for the system. The requirements are also based on the study of the existing systems in Part II. This leads to the final specification of the system. Based on the specification, a design of the system is presented. The architecture and access control system is developed. Next, we suggest existing systems for reuse. Implementation aspects are discussed and we give suggestions for future work.

Part IV Conclusion

Here we sum up the results of the thesis work.

Part V Reference Section

This section contains the referenced literature and the appendixes.

2 Background

2.1 The Needs to Publish

Scientific staff at universities and university colleges in Norway are experiencing a stronger push towards publishing in international scientific journals. The number of publications influences the chances of employment and job advancement. For those working with research where the product is not a scientific report, but rather a more abstract of physical product, the pressure towards publishing may be hard to live up to. The product from musical research work may for instance include musical examples, or may in fact be the music or sounds on a CD. To publish this for the public, the CD needs to be produced and distributed. The cost of making even a small stock of CD's may be quite high, and the costs increase for each CD made. Music published in conjunction with research is also most probably not mainstream enough to generate a lot of sales, so the project then has to rely on external financing to be able to publish the results of the research. The scientist may then use a lot of time and efforts just to get the publication or production financed. If, on the other hand, the publishing had been a report in an international journal, the scientist could expect to get paid a reward. From this it is straightforward to deduce the need for a cheaper way to publish sound material.

For these publications a solution may be to publish them on the net. Even though there is still the costs of recording av producing the music, this would save the costs of manufacturing the CD's.

2.2 Different Genres

People working with different genres of music will have different needs. To limit the scope of this project the main focus will be on covering some of the needs of researchers in musicology, and also to be a system for musicians and students of classical music. It will also be looked into the needs of the Norwegian traditional music archives to see if their publishing goals are similar to those of the classical musicology.

2.2.1 Classical Music

For performers of classical music it is important to be able to find the scores of the music you want to perform. This can be a time consuming process where you have to contact libraries or archives. A vast amount of music have been composed and published during history, but the very special composition that the performer is looking for may be very hard to find. Some music have only been published in limited numbers, and may simply be sold out and out of print. In reality it may only be accessible for a limited number of people with special access to musical archives.

In Norwegian Classical music there are some names that are very famous and their music is widely spread and used. Their music is well known, and not to hard to get a hold of. Music made by other and less known composers may be in no use today because so few people are aware of their work. For instance in the National Romanticism era a few composers like Edward Grieg and Ole Bull are well-known. Today we may see different qualities in music composed in times where it did not meet the common standards.

To be able to find the correct music it is important to know what music exists and where to obtain it. One solution to increase the accessibility of old scores is to make reprints of the music. This solution has some drawbacks. First it is costly to make and distribute printed material. Some of the material may also need to be edited and touched up before it is published. Second some of the same problems of availability remains. The material still has to be made known to the public, and is must be ways to search for and order the material.

A combined solution with information about composers, their compositions and access to the scores through one readily accessible place would meet some demands of performers of classical music.

2.2.2 Norwegian Traditional Music

This chapter presents some needs of Norwegian traditional music archives. It is looked into if this thesis work may be angled towards their goals and demands[1].

Today there are three main archives for traditional music in Norway[2]. In addition to this there are local archives all over Norway, dealing with the local traditions. The archives contains big collections of published and unpublished material. The focus areas are to take care of the collected material, to update the collections with new material and to make the collections available to researchers and the public. The archives contains among other things material in the form of transcriptions of dance music and songs, recordings with players, singers and informants and also video of dance and music. The archives works with preserving this for the future and make it accessible.

The people working at the archives spend a lot of time handling requests for archive material. A request for material may include samples from many different recordings. Even though the material is registered, the archive worker may spend a lot of time finding the correct tapes or disks. It is quite simply very time consuming to make copies of the material for the users. Ideally, all the material should be stored in a digital system with easy mechanisms for making the requested material available to people requesting it. Most of the archives plan to make their archives digital in the future, but this is easier said than done. The two main winnings of making the archives digital are that it is far much more easier to make backup systems to conserve material, and second that digitizing it makes it easier to make it available to research and the public.

1 The source for this chapter are lectures in Norwegian Traditional Music at Høgskolen i Telemark (2002-2004) and an seminar for Archives at Dragvol, Trondheim, March 2006.
2 Norsk Folkemusikksamling, Rådet for folkemusikk og folkedans (RFF) and The Arne Bjørndal Collection.

There are three steps that have to be completed before the archives can have digital distribution of the material:

1. **Preserve the Material by Making it Digital.**
   A lot of the material is currently stored on analogue, degradable media. It is important to transfer this into a suitable digital form before the material is degraded. Material such as a recording consists of the recorded sound and the sound bearer (tape or disc etc.) It is then the sound, not the sound bearer that can be preserved by digitization.
2. **Cataloging the Material.**
   The proper meta information about the material must be registered. All recordings and other material must be registered (down to individual track level) to make it possible to perform searches in the material.
3. **Distributing**
   The third step is the distribution of the collections. When the material is digitized and cataloged, it is possible to make it available in web-based solution. However, rights management issues must also be handled before it can be distributed.

The archives are currently focused on the first two steps. There are several obstacles to overcome before they will be able to complete the third step. The main obstacle seems to be that limited financial circumstances makes the process very slow. The material has been recorded on many different types of media, all of which require different player units, and possibly some audio processing. The recordings must be digitized in a way that preserves the signal quality of the recording. Nobody wants a situation where in a few years all the recordings have to be digitized a second time because better audio equipment is available. It must therefore be done properly the first time. This requires high technical skills and access to good audio equipment, and is in essence an economical question.

Another problem with the recordings is handling the intellectual property rights of the material. Not all the material can be made public since it may have been recorded under the assumption that it is not going to be published. Maybe the recorded player or singer was old and did not perform as well as he did when he was younger. Then he may not like the idea that the recording is spread to the whole world. To get copies of such material today it is necessary to get permission from either the player or singer or, if he or she is deceased, from their heirs. This will still be the case when the material is in digital form, and a digital system must support this.

The current software used for cataloging by many of the Scandinavian folk music archives is the Fiol® software system. It has been developed as a joint effort of Norwegian and Swedish archives, and the software development is currently maintained by Folkmusikens Hus in Rättvik in Sweden. The software runs on PC and Mac. In the Fiol®-database every medium used in typical archives (tape, video, CD, reel-to-reel, etc) can be registered, and the contents of the medium can be registered. A DAT-cassette may for instance hold many different tunes and interviews. All these can be registered individually. The latest release of the software includes the possibility to link the cataloged item directly to the digital media representing the track. This enables the archive to give archiving services to without wearing out the original media. They are also developing a application for use with Fiol® that enables the archives to connect their database to the web for the public to search through them. Fiol® can then link directly to files on a streaming server [1].

This chapter has shown that the needs of the folk music archives perhaps is not to develop new systems for distributing their archive material, but first and foremost to digitize, catalog and preserve all the archive material. The next step will be to make this available for others, but this brings with it questions concerning right management that still needs to be resolved. In light of this the folk music archive community has not been further involved in this project.

2.3 Needs Based on Interviews

The main source for input on the design of the system has been Harald Bjørkøy at the Grieg Academy in Bergen and Leif Arne Rønningen at NTNU in Trondheim. An interview with a potential user of the system has been done with Liv Mari Schei. She is a student majoring in musicology at NTNU with song and piano as main instruments. This chapter lists needs that has come out of interviews and conversations with Bjørkøy, Rønningen and Schei.

2.3.1 Harald Bjørkøy

Harald Bjørkøy is one of the proposers of this thesis. He outlines the following as elements for a on-line digital music archive system:

It should be an on-line service for storing and publishing sheet music, recorded music and information about the music. The system should be targeted for musicology high schools and universities, and function both as a “record company” and a sound and sheet music archive.

The system is first and foremost intended for researchers and specially authorized people at a University that wants to share information and ideas. In the future is should be possible to link the systems together to form a system that spans several universities in several countries. The system should enable the publishing of music instead of having to publish music through normal record labels.

All users of the system should be allowed to publish their work. Some of the material can be made available to all users, while other material is only available to special persons or groups of persons. Some of the material could be available in a way where some users have to pay to get access. An example is a research work ending in the recording of a piece of music. Then the music could be available for free to other researchers in the same University, while other users have to pay a fee of for instance 1€ to download it. It could be payment solutions for part of the material in a similar way as normal web-based music stores.

There should be mechanisms to keep track of the quality of the published material. The published material should be of a certain quality. There could be different measures of quality, for instance a quality scale where the highest level is top level research, and the lowest level is home-made material made by amateurs.

The system could also support conference groups for instance between scientists or researchers in working the same field. This should make it easier to cooperate in projects and to share information.

2.3.2 Leif Arne Rønningen

To avoid listing the same needs once more, only additional or differing needs are listed.

Recorded music must be available for other users to listen to or download. Video recordings should also be handled in a similar way. For sheet music it is preferable if there is a easy way to listen to the scores.

Artists or groups could use the system to publish music. Either for free to promote themselves, or the music could be available in a web-shop style.

There could be a automatically generated list of "best-sellers" that lists the most accessed material or the material that have lots of downloads. The system could show the latest released material. This could be sorted by category. Users could recommend material by logging on to the system and comment, vote for or give an analysis of the material. The system could generate lists of the most popular material based on these votes, and other users can use these lists to decide what to buy or download.

2.3.3 Liv Mari Schei

The system should have information about different composers. It could be like an encyclopedia where both information about composers and their compositions is available. It should also be easy to get hold of the music or information on where it is found. She has not felt the need for such a database earlier, but she thinks it could be very useful as long as it is made known to the potential users.

Part II Existing Systems

3 Evaluation of Representation Formats

In this chapter some existing file and representation systems are evaluated to see if they can be used as a basis for building the system. Finding a suitable representation form for sheet music has been emphasized due to the fact that digital sheet music archives are less common and less described in literature than formats for audio and video.

3.1 Media formats

The system is going to handle different types of media. These are audio, sheet music, video and information about them.

3.1.1 Representing Music

There are several ways to represent music. [2] presents some possibilities:

- Printed notation
- Human-produced MIDI[3] file
- Audio replayed from a human-produced MIDI file
- Audio synthesized from an internal representation
- Audio file representing a human performance
- Internal representation, suitable for searching

It is not easy to generate an internal representation from a human audio performance. It is also not possible to make a computer synthesize a human representation, although there are attempts to make algorithms that generate "humanized" audio. Converting between the other representations is possible but with a varying degree of success regarding the result.

3.1.2 Digital Audio

For storing audio one should strive to keep the audio at a best possible quality. Digital audio is characterized by the number of samples per second (sample rate), and the number of bits used per sample (word length). For audio, IASA, the International Association of Sound and Audiovisual Archives, recommends the following [3][4] :

Sampling Rate:	48kHz minimum, 96kHz preferred
Bit Depth:	24 bit minimum

3 See section 3.3.1

If the original audio is on a digital source as a CD[4], the audio should be stored at the existing bit depth. Increasing the sample/bit-rate will only increase the space occupied by the file. If audio is recorded especially for the archive one should strive for higher sample/bit-rates, since this lay fewer bands on how the audio can be processed at a later stage. As an example, say that the audio is going to be manipulated in an application where it is stretched in the time dimension. A high sample rate on the original signal allows more stretching before the human ear can hear that the signal is distorted. Artifacts as clicks and noise in the original recording should also be treated as part of the sound signal. They should not be removed from the digitized object to produce a better signal since that makes it impossible to perform a better signal restoration in the future. Digitization of music or sound stored on analogue media can be a challenging task which requires advanced and often expensive equipment. It is beyond the scope of this thesis to describe guidelines for the digitizing process in detail. More information on the topic can be found in [3] and [4].

There are many file formats for sound available. IASA stresses that the coding of audio for audiovisual archives must be based on open standards and not proprietary formats [3].

It can be separated between compressed and uncompressed files. Compressed sound files can be further grouped into lossy and lossless compression. With lossy compression the reconstructed signal is not identical to the original signal, and information is lost. A file compressed with lossless compression can as the name imply be completely and perfectly reconstructed. Figure 1 shows this split up of sound file formats, including examples of well-known file formats. The list is not supposed to be a complete listing of sound formats, it is just meant to provide an overview of some of the available technologies.

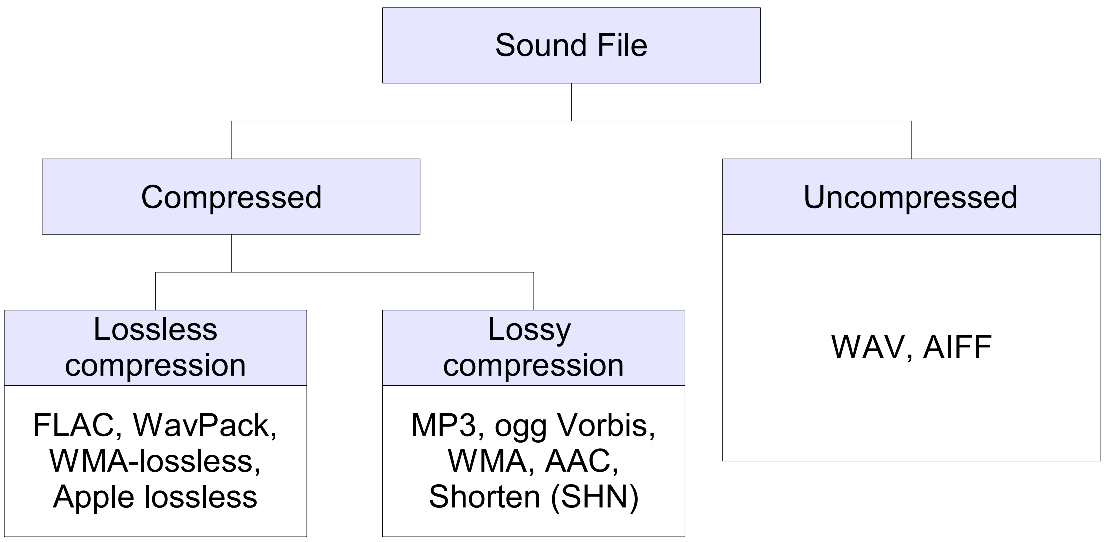

Figure 1: Some of the most used audio formats for sound split into lossy and lossless compression and formats with no compression.

4 Often called “born digital”

3.1.2.1 Uncompressed audio

Uncompressed audio stores all information from the audio samples. Computer programs normally work with uncompressed audio internally, although they can read and save compressed audio. PCM, or Pulse Code Modulation, is a way of representing an analog signal digitally where the analog signal is sampled regularly at uniform intervals. PCM is used for CD-audio as well as the internal format in many computer systems[5]. Normal CD-audio is sampled at 44.1 kHz at 16 bit, which mean that the audio signal is sampled 44 100 times per second, each sample being 16 bit. The human hearing normally works in the range between 20 Hz and 20 kHz. According to the Nyquist theorem this signal has to be sampled with the double frequency (2*20Hz = 40 kHz) to be represented perfectly. CD-audio uses 44.1 kHz which allows the audio filters to be imperfect.[5]

Wave files (*.wav) is a sound format originally made for Windows machines by Microsoft. Due to the success of Windows and the amount of programs written for the platform, the wave file format is supported by most modern music programs and platforms. Different sample rates, number of channels and bits per sample can be specified. Wave files can be compressed, but the most used format for wave files is the uncompressed PCM-format. For two track recordings IASA [4] recommends wave files (.wav) or better, Broadcast Wave Format (.bwv) files. BWV is a file format developed by the European Broadcasting Union based on WAV with extra metadata information.

3.1.2.2 Lossless compression

Lossless audio compression algorithms exploits the fact that sound files usually contain redundant information to shrink the files. The bit information in the files can be completely restored. This is similar to the way ZIP-compression works. The difference is that the algorithms are optimized for the way audio and the files can be used directly in audio programs without unpacking. Lossless audio compression can normally compress the file size down to 50-60% of the original size[6], which may save a considerable amount of space on a file server. However, there is more to consider than just the file size of the compressed audio. The speed and CPU load of the encoding and decoding algorithms must also be taken into account. If audio shall be streamed to users, the audio format should also support streaming.

There are many lossless compression algorithms available. One much used format is FLAC which stands for Free Lossless Audio Coding. The algorithm is optimized to be faster to decode than to encode, and the decoding only require integer arithmetic. FLAC is widely supported by audio hardware and software on most operating systems. It is streamable and seekable[6], which means it can be handled in much the same manner as uncompressed audio [7].

3.1.2.3 Lossy compression

Lossy compression[7] uses psychoacoustic techniques to remove information from the original signal that the human ear normally would not hear anyway. This is combined with

5 PCM is not a file format
6 FLAC support sample accurate file seeking.
7 Some use the term “reduction” instead of “compression” since the information is reduced

compression methods that removes redundancy from the file, and very good compression is possible. With standard 128 kbit/sec MP3-encoding the file size is about 11 times smaller than the original CD-audio (44100/128 ≈ 11) and still the normal user might not hear the difference. To compress the signal it is possible to reduce the sample rate to reduce the number of bits per sample. Reducing the sampling rate results in reducing the frequency resolution. Most compression routines instead concentrate on reducing the number of bits used per sample. As mentioned, CD-audio uses 16 bits per sample. This is to get a good signal-to-noise-ratio (S/N). Every bit gives a 6 dB[8] increase in S/N[8]. If the number of bits is reduced you would experience more noise, this is most audible when the general sound level is low. When the sound level is high, the noise level can be increased because noise will be masked by the human hearing due to the way the ear and the hearing works. This explains why this bit reduction technique compresses the signal while the perceived quality remains the same as the original. The last step in the process is removing redundancy by using Huffman[9]-coding of the signal, which is a lossless coding technique[6]. Other lossy audio file formats include vorbis (file extension: OGG), AAC and lossy Windows Media Audio (file extension: WMA).

Since lossy compression makes permanent changes to the audio it should not be used for primary storing. An unreduced original should be used as the master copy. The benefits of the compression can be exploited when sound is to be streamed or downloaded to users of the archive system since it will use less network resources and less storage space in the users machine.

3.1.3 Video

The system is supposed to handle video. There are many different formats for video, both in the analogue and the digital world. It is beyond the scope of this thesis to go into detail about all the analogue and digital formats, but the reader should be aware that there have been and still are many different formats for video and TV. The major differences are:

- The sampling rate
- Interlaced versus non-interlaced frames
- Image aspect ratio (vertical and horizontal size)
- Color coding
- Audio coding
- Gamma correction due to non-linear relationship between signal voltage and light intensity
- Compression routines (digital video)

There are three major TV-systems, each with its own twist on how to code and display the signal[9].

8 The decibel is a logarithmic unit often used in acoustics to quantify sound levels relative to some 0 dB reference.
9 Huffman variable length entropy coding

- NTSC (National Television Standards Committee) – used in North-America and Japan.
- PAL (Phase Alternating Line) – used almost anywhere (including Norway)
- SECAM (Sequential Color Memory) – France and parts of Greece, Russia, Eastern Europe, Africa and some other places.

Some important differences are listed in table 1. The interested reader could find more information in [9].

Parameter	NTSC	PAL	SECAM
Field rate (Hz)	59.94	50	50
Lines per frame	525	625	625
Gamma	2.2	2.8	2.8
Audio carrier(MHz)	4.5	QAM	FM
Color modulation method	QAM	QAM	FM

Table 1: Properties of the three major television standards of the world.

(source: [9])

In addition to the 50+ analogue video formats there are 15+ digital formats[10]. Analogue formats include Betacam, VHS, Hi-8, S-VHS, Betamax, V2000 and U-Matic. Digital formats include Digi-Beta, Beta SX, IMX, miniDV, Digi8, DVCam and DVC-Pro[10]. Converting video to digital formats for use in a digital library may be both costly, time-consuming and technically challenging. It is outside the scope of this thesis to go into details of procedures and guidelines for digitizing video.

3.1.4 Digital Video - Storing and Presentation Formats

There are many different formats for storing and streaming digital video. As for audio there is the choice of whether or not to compress the data. Compression should not be used for the preservation of analogue footage because it creates artifacts which affects post-processing of the video. The information in the original footage will then be forever lost when the analogue master medium sooner or later starts to deteriorate. A digital master copy using no (or lossless) compression should be made. From this master copy it is possible to make user copies that are better suited to streaming, downloading or browsing[11].

For archiving video a file format called MXF (Metadata eXchange Format) is proposed as a future standard. It is a container format for multimedia, for instance video. It can hold many different types of video like MPEG, DV and uncompressed video in addition to other types of data and metadata. It is Developed by the Pro-MPEG Forum and the G-FORS Group and standardized by SMPTE and EBU. The format supports the whole production chain from filming to transmission to storage in archives. Metadata about the different steps of the production process may be stored as metadata in the same file as the data itself [10]. Although MXF is getting increasing support by manufacturers, vendors and users, interoperability issues have been raised. According to [12] two major camera systems[10] for broadcasting which produce MXF output make files that are incompatible with each other and impossible to tell apart without the use of advanced tools.

For preservation of video the NISO Framework Advisory Group recommends the following:

> *High resolution video files are huge and digital file formats for preservation quality video are immature. Therefore, at this time most organizations maintain their best archival copies of video content in media-dependent form. Preferred media-dependent formats contain a minimally compressed or uncompressed signal, e.g., DigiBeta, D1, or D5 tape. In a high bandwidth LAN, access copies may be high-bit rate MPEG-2 or MPEG-4 files in larger picture sizes; for lower bandwidth applications and the Web, one may present lower-bit rate MPEG-4, RealVideo, or QuickTime formats with smaller picture sizes.[11]*

Examples of video formats in much use today are MPEG-1, MPEG-2, WMV and AVI. These are all container formats, meaning that they are containers for interleaving audio and video streams (and possibly other streams too). A container format can contain video compressed with many different codecs or plain, uncompressed video. Codec is software that can encode and decode digital data like a video or an audio stream. The word codec is a term that means "compressor/decompressor" or "compression/decompression algorithm". The video file format does not necessarily imply which video and audio codec is in use, and vice versa. To watch video on a computer, the correct codec must be installed otherwise the playback software will not know how to interpret the data. A user might therefore experience that watching one AVI film works perfectly, but when he opens the second AVI film, only the audio can be heard. The user is then missing the second video codec, although he has the correct audio codec. If he installs the second video codec, he will be able to watch that movie too.

[13] suggests five main issues that separates the container formats:

1. Popularity. Widely used formats may continue to be much used although they are less advanced and produce bigger files.
2. Overhead. The container formats may have different file-sizes although they have the same content.
3. Support for advanced codec functionality. (B-frames[11], VBR[12], etc.)
4. Support for advanced content. (chapters, subtitles, meta-tags, user-data, synchronization information, etc.)

10 Panasonic DVCPRO P2 and Sony XDCAM
11 In MPEG compression, a B-frame only contains the data that have changed from the preceding frame or is different from the data in the very next frame.
12 Variabel Bit Rate, a coding scheme that assign more bits to complex parts of the signal and less bits to less complex parts.

5. Support of streaming media. In addition to being a format that can be stored and downloaded, a streamable format is usable since it allows video content to be viewed without downloading the whole file first.

Streaming video to users via the Internet is done with streaming software or hardware. Normally a dedicated streaming server is used. Which streaming server to use depends on factors as cost, ease of maintenance, performance and what formats one wish to stream. If a digital library is going to let users upload their own movies, one can expect that users wants to use many different formats. They might be unaware of differences between different video formats, and only wants to upload what their digital camera[13] has captured on film. Such considerations should be kept in mind when deciding what streaming system to use.

3.1.4.1 MPEG

The most used representation format for video is MPEG. MPEG is a set of international standards from the ISO Moving Picture Experts Group. The development of MPEG started as an attempt to make a digital format to represent video in VHS quality. The standard was to be open and have a target bitrate of 1,5 Mbit/second including both the video and the audio data, and should be playable from devices like CD-ROM[2].

MPEG-1, the first of the MPEG standards, became a success. It had pixel resolution of 352 x 240, color picture, audio in near CD-audio quality and a frame rate of 30 pictures per second. This was both within the 1,5 Mbit/second goal, and could be decoded in hardware with 512 Kbytes of memory. Other variations of MPEG-1 is possible but is less common. MPEG-1 defined different layers for the audio coding, so that the audio could be specifically coded to suit the purpose. The third layer, or MPEG-1 Layer 3, is the audio coding in the popular MP3 format [2].

MPEG-2 was designed to handle high quality video. It is an container for several ways to code video and audio. It expands MPEG-1 by adding support for interlaced video and more advanced audio coding schemes. It defines a transport stream which makes it suitable to be transported over unreliable media. MPEG-2 is used in both broadcast such as directs broadcast satellite and cable TV. It is also with some enhancements the coding used in HDTV applications. Decoders for MPEG-2 is fully backwards compatible with MPEG-1. The audio coding of MPEG-2 is also extended to support more than two channels. To use MPEG-2 licensing fees must be payed to the Moving Pictures Experts Group [2][14].

MPEG-4 is not an evolution of MPEG-1 and 2, but another approach to multimedia contents. MPEG-4 is based on objects instead of only synthetic streams of video and audio. Together the objects form a multimedia scene. The different objects are transported in different streams. Objects can be natural like an audio recording or they may be synthetic such as 3-D polygon models. A movie could for instance be put together of four objects: audio, video, subtitles and titles. The coding of the objects can be individually adjusted to minimize the file size. A MPEG-4 movie could also have two video objects, where the first is intended for hand-held, low bitrate devices with small screens, and the second video stream is for high resolution display.

13 Digital cameras often have the ability to record movies in a format which the user can not change.

MPEG-4 is an advanced standard, and most implementations use only parts of the standard. MPEG-4 defines profiles and levels so that applications can communicate which functionality and decoder capabilities it implements. MPEG-4 can mean both the container format, the file format (.mp4) or the audio, video and other codecs defined in the standard. The standard has a huge potential for both video and audio in many areas of application. One potential drawback is that license fees must be payed to the Moving Pictures Experts Group [2][15].

3.1.4.2 Windows Media Video 9

Windows Media Video (WMV) is a set of video codecs developed by Microsoft. Like the MPEG standard, WMV exists in different versions, targeted at different areas of application. WMA version 9 has been standardized as SMPTE[14] standard 421M , and is not a proprietary format like the rest of the MWV formats[16]. The SMPTE standard 421M, commonly called VC-1, is a standard for high definition video that can be used to show DVD standard movies on personal computers.

Windows Media Video can be bundled with Windows Media Audio into the following containers [17]:

- Advanced Systems Format (ASF) container, a Microsoft format which can support DRM. File extension: .wmv or .asf
- Advanced Video interleave (AVI) container. File extension: .avi
- The Matroska container. File extension: .mkv

The WMV format is one of the most popular formats on the Internet for distributing video[17].

3.1.4.3 QuickTime

QuickTime by Apple has evolved from being a video format to be a whole environment for programming and displaying multimedia content. It consist of three parts[18]:

1. the QuickTime file format (File extension: .mov .qt and .moov)
2. a Media player for QuickTime files
3. a software developing kit for Windows and Mac

The QuickTime format exists in many versions, and the relationship between MPEG-4 and the latest versions of QuickTime is close, in fact MPEG-4 Part 14 is based on QuickTime [2][18]. One QuickTime file is a container that holds one or more tracks. Each track may be some kind of data like audio, video or text. The track may either be the media stream or a pointer to data stored in another file or network place. The QuickTime format is well suited for editing since it is possible to edit multimedia files without changing the multimedia data. Only the pointers in the QuickTime file is changed, and the streams pointed to by the tracks need not be re-encoded. It is said that the QuickTime format support a bigger part of the life-

14 SMTPE is an acronym for Society of Motion Picture and Television Engineers

cycle of a multimedia presentation since it supports creation, storing, editing and streaming of content [18].

3.1.4.4 RealMedia

RealMedia is a container format for RealAudio and RealVideo, developed by RealNetworks, Inc. RealAudio and RealVideo are compressed audio and video formats intended to be used for streaming. The latest version of RealVideo, RealVideo version 10, uses a proprietary video codec called RV40. RealMedia can be streamed using the Real Time Streaming Protocol (RTSP) for playback control. The video stream is sent via RDT, a proprietary protocol owned by RealNetworks. Video is normally streamed with constant bitrate, which makes it easier to predict the network use when streaming. Recently, support for variable bitrate has been included in the standard. Because it streams using its own streaming format, it is not straightforward to play RealMedia streams using other player than the official RealPlayer [19]. However, RealNetworks claims that RealVideo version 10 offers substantially better compression for the same video quality compared with other well known formats such as MPEG-4 and WMV-9 [20].

3.1.4.5 AVI

AVI is an abbreviation for Audio Video Interleave. As the name implies, it is a container format that can interleave audio and video. The format traces back to Windows 3.1 and the development is now obsolete. It is not as technically advanced as some of the other container formats, with no native support for advanced features like B-frames, VBR-audio and Variable Frame Rate(VFR). It supports video and audio in most compression formats, although some through “hacks” that create incompatibility problems. It is widely used in file-sharing communities, although the overhead when used with modern coding schemes makes the files unnecessary large [21].

3.1.4.6 Bitrates

Typical bit rates of DVD-quality video for modern codecs is typically around 750 kilobits per second [15]:

Format	Bitrates (approximations)
MPEG-4:	1000-1500 kbit/s
MPEG-4 AVC:	750-1000 kbit/s
Windows Media Video 9:	750 kbit/s
MPEG-2:	6-8 Mbit/s
MPEG-1:	1.2 Mbit/s for VHS-quality!!

Table 2: Typical bitrate performance for popular video formats.

3.1.4.7 Streaming Servers

A streaming server is used to stream video and/or audio to users. Table 1 shows three different video streaming servers from three vendors that are commonly used for streaming (table data from [22]):

Name and Vendor	Cost	Control of standard	Type of media streams
Apple QuickTime/Darwin Streaming Server	Free	Open source. Open standards	QuickTime and MPEG-4 content
RealNetworks Helix Universal Server	Expensive	Proprietary	Real, Windows Media, MPEG, and QuickTime content
Microsoft Windows Media Services	Free with Windows Server	Proprietary	Windows Media content

Table 3: Comparison of three common streaming media servers

3.2 Possibilities of Digital Musical Scores

To decide the best format for storing digital sheet music in an archive it must first be clear what the sheet music shall be used for. It can range from simply watch the sheet on screen to let the user download and edit the scores. The following is list of things that users might want to use the sheet music for (not listed in order of importance):

- Watch the music on screen
- Print the music on paper
- Listen to the music, maybe with the possibility to adjust tempo and key
- Edit/manipulate the score (directly or by downloading the source file)
- Use sheet music as part of an on-line document, possibly in collaboration with others

3.2.1 Display, Print and Listen to the Scores

The first thing is to display the sheet on screen. This can be done by storing them as image files (TIFF/JPEG/GIF) or as PDF files. The drawback of using a PDF document for display is that the user need to have a PDF document reader (like Acrobat Reader) installed to view the score, although this is a de facto standard for document exchange and is very common. For sheet music that spans multiple pages using PDF can be a good idea because the score can be stored on the server as one file instead of breaking it up into several image files. Printing scores from music notation programs often support printing to PDF directly or through a PDF-writer plug-in, which makes it fairly easy for users to make PDF files.

In addition to letting the user watch the sheet music on screen, it will probably be a need for printing it on paper. PDF documents will be a good choice for this, as all formatting is kept like the original irrespective of which program or computer system was used to scan or make the score. More guidelines to the digital capture of musical scores can be found in an excellent article by Jenn Riley and Ichiro Fujinaga[23].

A sheet music archive with the possibility to listen to the scores gives the user more ways of choosing and searching. For instance it gives the user a faster way of deciding which score to purchase and download. To record all the sheet music with professional musicians would probably be overkill, so a simpler approach should be chosen. The point is that the user would want the general idea of the composition and how it sounds. Standard home computers are equipped with sound cards and programs that plays standard audio files like MIDI. The upside of using MIDI is that almost all music software can import MIDI files, the file size is very low and MIDI files can be automatically generated from many score notation formats and programs. Most computer systems can play MIDI-files, and web browsers commonly support playing MIDI. On the other hand it is entirely up to the software or hardware that plays the MIDI-file how it sound. (see 3.3.1). MIDI may sound entirely different on a professional synthesizer than on a standard home computer. None the less it will give an impression of how a work of music might sound. Technology to make the playback of computer generated music sound more like human performers is called “human playback”-technology. This could be combined with MIDI to make the user experience even better.

3.2.2 All-in-one Display, Play and Print Alternatives

One alternative is to have the sheet music stored in a file format for digital scores and have server or client software that can display, play and print the score. Several solutions exists, both as plug-ins to the web browser, as programs run in the web browser or as stand-alone applications.

One example of such is a web-browser plugin called Scorch. When installed in the browser it can show sheet music on web pages along with the possibility to turn pages, change playback device, control playback and tempo, change key, save and print the score [24]. It is also possible to only let the user play an excerpt of the score without the possibility of saving and printing. This can be used to support previews. The plug-in is free to download and use, but a program that can make Sibelius .sib-files is needed such as Sibelius or Photoscore. The full versions of these programs might be to costly for the private user, though the ease of use makes it an alternative worth considering. Sibelius can export scores directly to web pages, which makes it very easy to use for people with little computer knowledge. The system serves at least as a good example on how sheet music can be made available for web. On a web-page the score will have controls mounted on top of the score as seen in Figure 2.

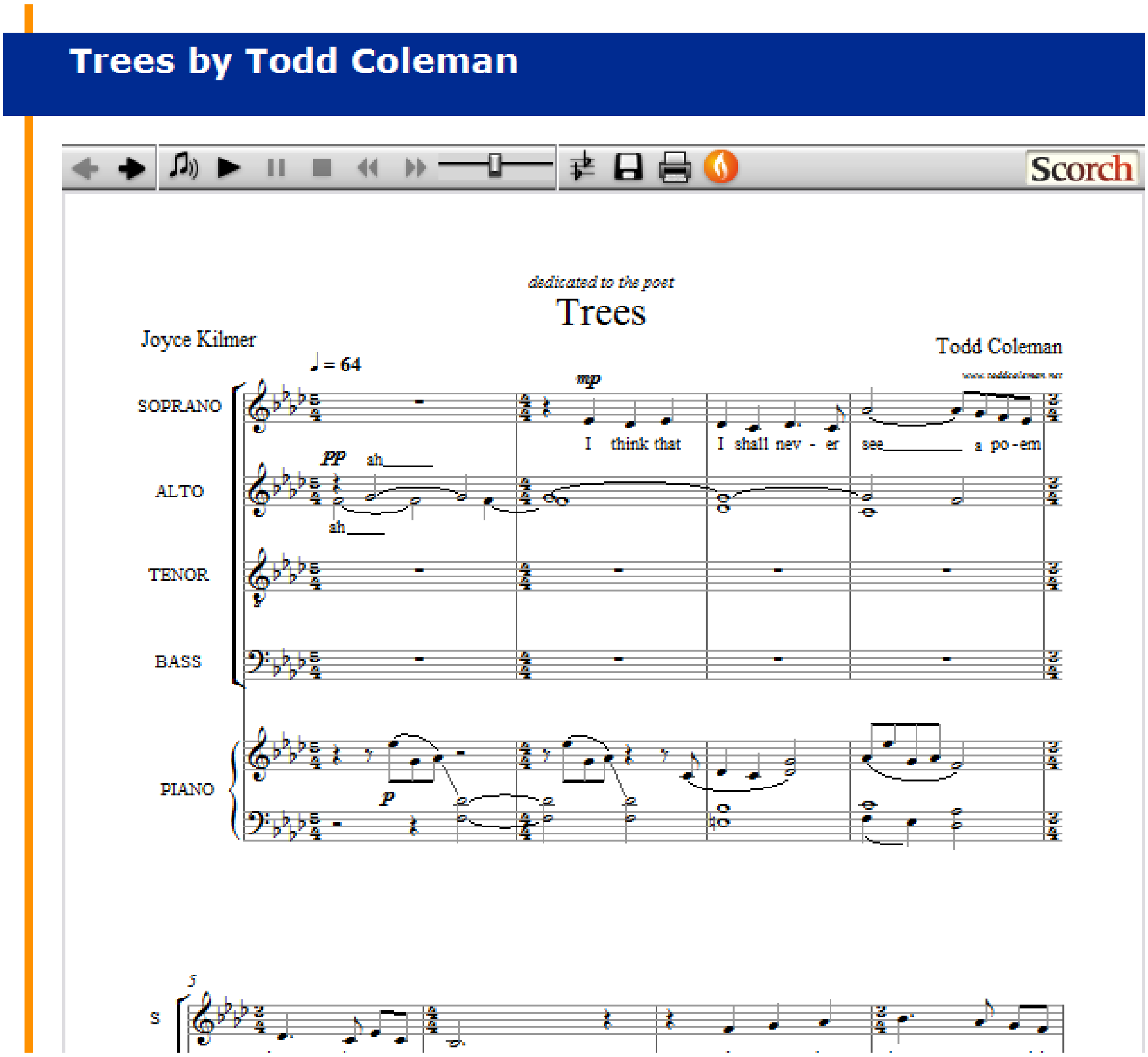

Figure 2: The Scorch web browser plug-in is shown displaying a playable score for choir and piano. Above the score controls for playback, save, tempo and pitch change and printing is found.

A similar system to Scorch that do not require installation of a non-standard plug-in is musicRAIN[15] which require the more standard Macromedia Flash plug-in. MusicRain uses MusicXML as the file input format and the look and feel is generally the same as for Scorch. More information about the MusicXML file format can be found in chapter 3.3.5. Figure 3 show the user interface of musicRAIN 2.0 with the controls for play, pause, stop, volume and key change, metronome, page turns etc. At the time of writing no price information was found about musicRAIN. More information can be found in [25].

15 See http://mediarain.com/musicrain/demo.php for a demo of musicRAIN 2.0

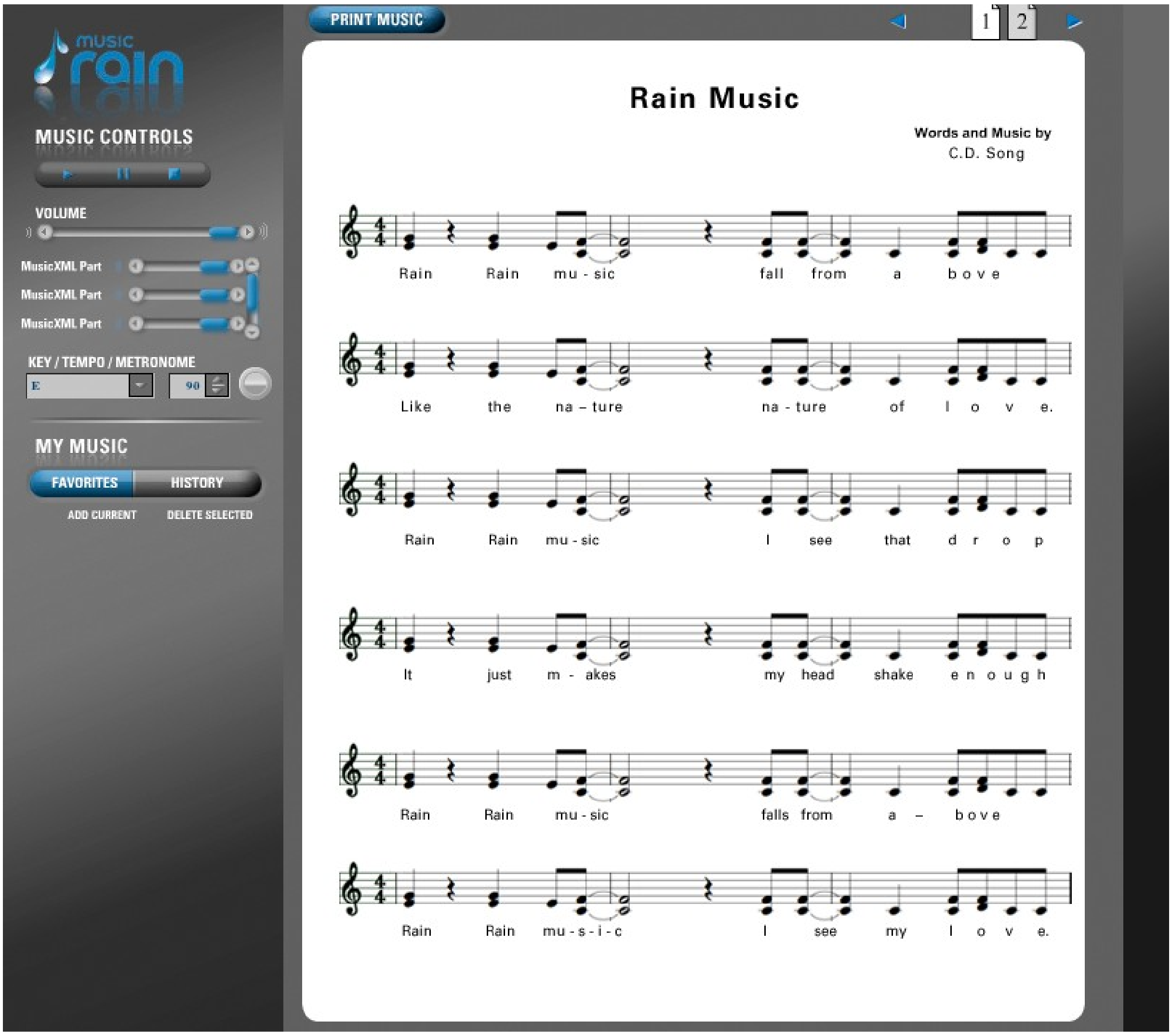

Figure 3: MusicRain 2.0 user interface with controls for i.a. volume, playback and key change

3.2.3 Edit/manipulate the score

If the users shall be able to edit and manipulate the scores themselves the sheets have to be able to be opened in the users notation program or edited directly in the browser. The experienced musician will probably want open the score a notation program he already knows instead of having to learn a new program. The list of notation programs or systems is quite long, and offering all file types for all the programs is a lot of job. Section 3.3 describes different file formats for storing music notation and looks at ways to deal with the variety of different programs and file types.

3.2.4 Use sheet music in articles

In an article about a musical topic, the writer might want to include scores as examples inside the text. This can be done by including the scores as images, but that makes the scores difficult to edit without including the source files. An example of a system where musical score notation can be included directly into web documents is WikiTeX, a mediawiki interface that support editing musical notation in the LilyPond language directly in wiki articles[16]. Software to include LilyPond score notation into LaTEX and HTML documents also exists, see 3.3.4.

3.3 Notation Language for Sheet Music

Different file formats for storing sheet music exist, both proprietary and free formats. To of the marked leading notation programs, Finale and Sibelius, each use their own formats. Finale uses MUS and Sibelius files are called SIB. These are not open standards and are subject to constant change. For example, Finale 2005 can open files saved in Finale 2004, but Finale 2004 can not open files saved in Finale 2005. If Finale files are to be offered, offering only the Finale 2005 version of the file format would hinder users with previous versions of the program from using the file. The file format chosen for storing the scores should have the following properties:

- Stable format that is not subject to constant change
- Free of charge
- Must store all notation information

The following sub chapters presents some possible file formats.

16 For more information on WikiTeX: see chapter 6.2.1.1

3.3.1 MIDI

MIDI stands for Musical Instrument Digital Interface, and is a protocol originally made to allow musical instruments to communicate in the early days of synthesizers. It has also become very popular as a interchange format for music audio. The problem with MIDI in our context is that it is basically a signaling language made to make musical equipment from different vendors work together. It can represent musical events like «note on» and «note off», which instrument is in use, and the volume and intensity (key velocity) of the keystroke. MIDI does not carry any sound information, it only signals what sound to be played and how. The file size of songs in MIDI-format is small compared to formats for exchanging audio. Because of the small file size, MIDI has been a great success for exchanging music on the Internet when the bandwidth was much more limited. It is possible to import MIDI-files into many music applications, but the result when converted to scores is not optimal and will vary according to choices made by the developers of the application. MIDI contains no information about chords, rests, cues etc. It also treats enharmonically different tones as the same tone, so the E sharp and F is the same tone in MIDI. The list of cons could be extended, but this should be sufficient to show that MIDI is not a good alternative for storing score information, and it was never intended to be.

Metrics for evaluating the different formats have been developed as part of the work with the thesis, and all the formats for score representation are compared using the same metrics. Table 1 lists some of the possibilities of the MIDI file format using the developed set of metrics.

MIDI	
Display score on screen	No, there is no one-to-one mapping of MIDI to graphical notation of sheet music
Print score on paper	No
Hear the score	Yes
Hear and watch the position in the score simultaneously	No
Let user user download and edit the sheet music	No, but the MIDI music can be edited with standard music software
Include score in articles	No
Software support	Widespread software and hardware support on most computer platforms. Almost any music program can import MIDI
Coding and readability	The MIDI data stream consists of a series of messages representing musical events (note on, note off, etc.) with timing information etc. [26]. Not easily readable by humans
Stable format	Yes
Free of charge	Yes
Store all notation information	No

Table 4: Summary of MIDI capabilities

3.3.2 Free Formats for Musical Score Representation

There are several free standards for expressing score music available. The main reason for choosing a free format is that there is no risk that some greedy company will start charging money for the use of the format. For a project with a possibly very long lifespan, it would be wise not to lock in to a proprietary format. One more reason not to use a proprietary format is the limited interoperability between different programs.

A lot of energy has been vested in making standard formats for common document types. One of the main reasons for the success of the World Wide Web is the standardization of the presentation format/language HTML (Hyper Text Markup Language), along with HTML's ability to link directly to other documents on the web (hyper linking). HTML made it possible to view the same document on a wide range of different operating systems and equipment. In it's basic form it consists of the text itself and tags which describes the basic layout. The details on how the layout is formated depends on the screen settings, the web browser and computer in use.

There are several formats for written text. A common, but proprietary format is Microsoft's Word-format. It's success is mainly due to the success of the Microsoft Office-package in both the corporate and private domain. Adobe PDF (Portable Document Format) is emerging as a common standard for exchanging documents across different platforms and systems. The documents can be both text, graphics, images and combinations of these. A PDF-document can be viewed and printed on any platform, still looking as the original document. Free document viewer programs and plug-ins for web-browsers are offered by Adobe.

In the next part we will be looking at free formats for score and music representation. Several formats exists, and they all have their pros and cons. It is not within the scope of this document to describe all formats. Three formats are described here, GUIDO, LilyPond and MusicXML. PDF is not recommended for storing notation because although it can be used to display scores, it treats the scores as graphics, which makes it awkward to use it for editing and listening.

All three formats are text-based. Text-based notation has several advantages:

- The format is not locked to a particular computer operating or software system.
- They may be typed and read by humans using only a simple text editor instead of an expensive commercial notation program.

It is for some of them fairly easy to construct software that for instance can generate music based on algorithms without an extensive computer knowledge.

A more comprehensive summary of notation languages can be found in the article “Musical notation codes” by Gerd Castan [27] and some formats are described in the book “Beyond MIDI: The Handbook of Musical Codes”[28].

3.3.3 GUIDO Music Notation Format

GUIDO is a formal notation format for scores and music. The format is plain-text, human readable and is platform independent. GUIDO can be used to notate both simple scores and advanced systems of scores such as symphonies. It is capable of representing all information normally found in sheet music.

The format is free and supposed to be very adaptable. It was developed with focus on adequate representation of music. The scores can be notated by hand with a ASCII compatible text editor. Simple musical concepts that is much used is easy to notate, while advanced and rare concepts have a more advanced notation. This is done so to make the format adequate for manual typing and writing. The format is developed with focus on musical representation in contrast to formats with a graphical notation as main focus.

GUIDO has a basic format that can be adjusted and extended to cover the needed notation. It is organized in three layers. These are Basic, Advanced and Extended.

- Basic GUIDO covers normal syntax and basic musical notation.
- Advanced GUIDO expands this with exact score formatting and more musical concepts.

- Extended GUIDO further cover concepts beyond normal notation.

The GUIDO Noteserver is a on-line server that converts GUDIO Notation to GIF or postscript The service is free and can be used to embed music in web pages as pictures or MIDI sound. There are four ways to include scores in web pages by use of the Noteserver[29]:

1. The GUIDO Noteserver can be used to generate a GIF picture or a MIDI file which is saved and included in the HTML-code.
2. An extended URL provided by the NoteServer can be used to get an automatically updated picture every time the page is loaded.
3. JAVAScript can be used in conjunction with the NoteServer, for instance if the score is stored in a separate file.
4. The NoteServer-JAVA-applet can be used for scrollable scores.

Next is a demonstration example of GUIDO notation with the corresponding image output from the GUIDO Noteserver in Figure 9.

```
{ [\title<"No.3"> \tempo<"Andantino">
    \staff<1> \clef<"g"> \key<+1> \meter<"3/8">
    \i<"p"> d2/8 |
   \sl(\dim(d h1)) h
   \sl(\dim(h g)) g
   \sl(\cresc(f# a c2))
   \sl(c h1) ],
  [ \staff<2> \clef<"g">  \key<+1> \meter<"3/8">
    _/8 |
    h1 _ _
    g _ h0
    \sl(c1 f# a)
\sl(a g)]

  }
```

Example of GUIDO Basic notation.

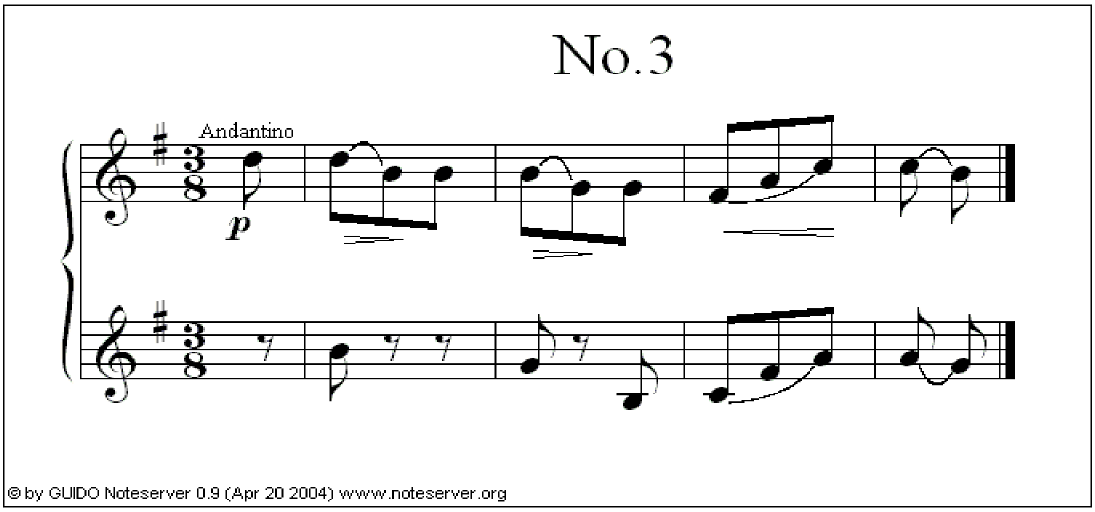

Figure 4: Score example generated by the GUIDO Noteserver

GUIDO is developed by the SALIERI Project. In addition to the Noteserver additional software is available for download from the SALIERI Project homepage [30]. One possible problem with GUIDO is that there seems to be little development on the project the last years and software for example for converting between Finale files is not provided for other versions than Finale 2000. The documentation for Advanced and Extended GUIDO could not be found. GUIDO is however used as the language format by a Gamera based sheet scanning project, see Chapter 4.2.2. It was chosen because of its ability to encode several levels of detail in a score, both visual and auditory details, so that the scores could be used in notation editing software [31]. Table 5 sums up the usability of the GUIDO format for use in a musical database.

GUIDO	
Display score on screen	Yes
Print score on paper	Yes. GIF images and EPS (Encapsulated Postscript) files.
Hear the score	Yes. GUIDO notation can be embedded in web pages and automatically converted to MIDI by linking to the Salieri Noteserver. A permanent MIDI file can also be generated from the GUIDO notation.
Hear and watch the position in the score	No, no software found.
Let user user download and edit the sheet music	Yes.
Include score in articles	Yes, GUIDO notation can be embedded in web pages and automatically converted by linking to the Salieri Noteserver.
Software support	Lack widespread support. Some plugins and libraries for C++ exists. Used as output format by a Gamera based sheet scanning project [31].
Coding and readability	Plain-text ASCII. Human readable.
Stable format	GUIDO Basic is stable
Free of charge	Yes
Store all notation information	Yes, because GUIDO can be extended.

Table 5: Summary of GUIDO capabilities

3.3.4 GNU LilyPond

LilyPond is free software program for text-based notation of scores. It is developed in reaction to the somewhat mechanical and sterile graphical output of notation programs, and aims at producing exceptionally nice score engravings. As well as being software for engravings it defines a free notation language and file format (.ly-files). The format has several things in common with GUIDO, for instance that scores can be written in plain-text by humans with no other tools than a ASCII text editor. The LilyPond software takes LilyPond notated files as input and compiles PDF, SVG[17] or MIDI files. The file format is supported for export by notation programs like NoteEdit and Rosegarden. LilyPond is operated from the command line, and does not have a graphical user interface. An example showing the use of LilyPond can be seen in Figure 5.

17 Scalable Vector Graphics

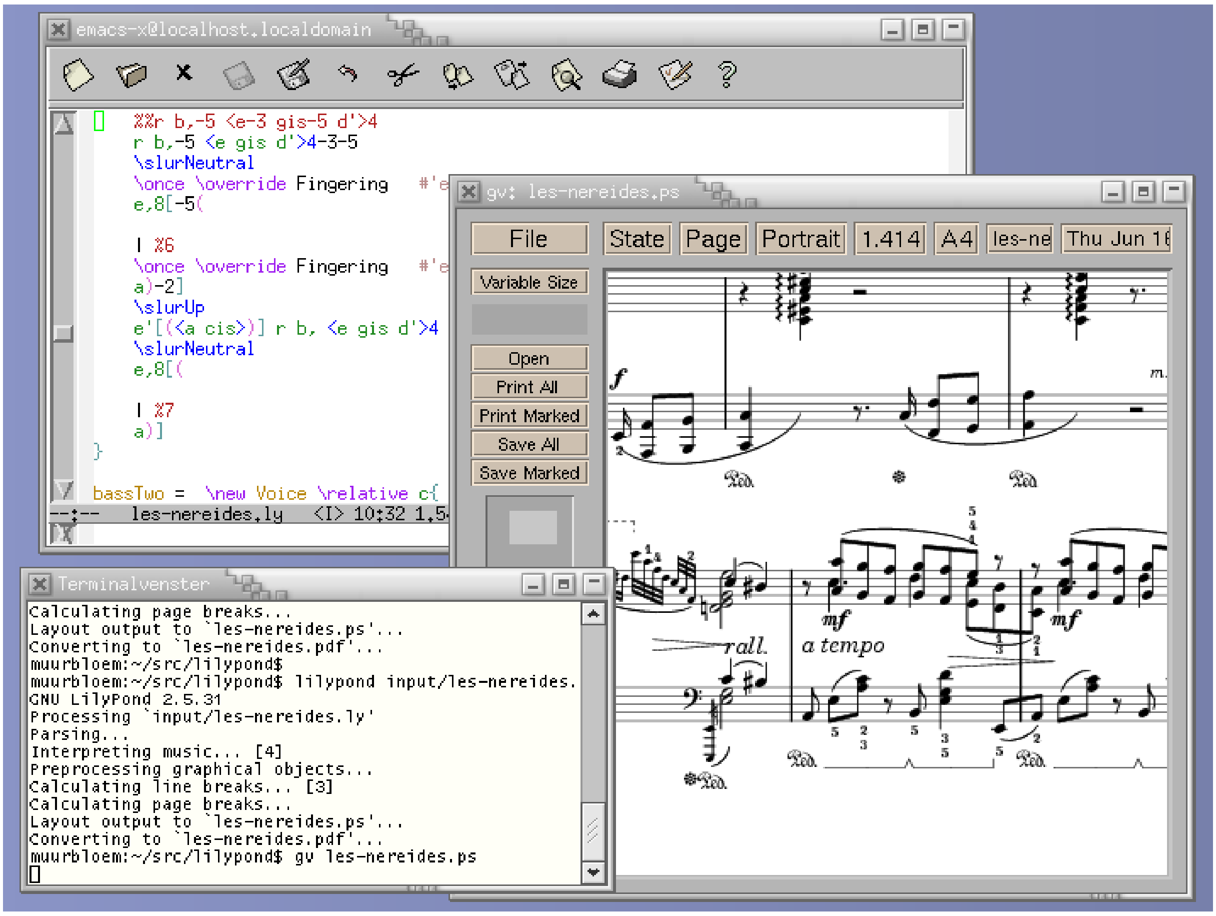

Figure 5: An example of the use of LilyPond. The upper command windows is used for text input, the right window shows the corresponding output as a .ps file.

LilyPond is in reality a compiler program that translates notation from text into graphics or sound. It can run in batch mode and can be scripted to convert multiple files automatically. This can be used by for instance musical databases. The Computer Science department at Utrecht University have used LilyPond to display scores on an on-line music database search system[32] and [33]. It is possible to compose algorithmically inside LilyPond using the Scheme programming language.

LilyPond can also be used to include musical notation into documents directly. The LilyPond software includes a tool called `lilypond book`, that takes the text document with LilyPond code as input and outputs the document with pictures instead of the music. LaTeX, `HTML` and `Texinfo` is supported. The music is included as tags which the software recognize automatically.

For HTML the music is entered using a tag called `lilypond` as seen below:

```
<lilypond fragment relative=2>
<\key c \minor c4 es g2
</lilypond>
```

`lilypond-book` then produces an HTML file with appropriate image tags for the music fragments as seen in Figure 6.

Figure 6: Picture generated by LilyPond

More examples of code and output can be found in [34]. Programs to convert to LilyPond from the following formats exists[35]:

- MIDI
- ETF (a Finale format)
- ABC (popular format for notating folk tunes etc.)
- MusicXML
- NWC (file format for NoteWorthy)
- Band In A Box files can be converted to LilyPond using Alain Brenzikofer's biabconverter

Table 6 sums up some of the capabilities of GNU LilyPond.

GNU LilyPond	
Display score on screen	Yes. Support for generating images for web.
Print score on paper	Yes. Support for automatic conversion to PDF and EPS format.
Hear the score	Yes. MIDI output is supported by the LilyPond software.
Hear and watch the position in the score simultaneously	No, no program found.
Let user download and edit the sheet music	Yes.
Include score in articles	Yes, support for LaTeX, HTML and Texinfo. With Wikitex, direct editing of scores notated in the LilyPond formats is possible in wiki articles.
Software support	LilyPond is software itself. It converts from LilyPond notation to PDF, EPS and MIDI. LilyPond output is supported by the Rosegarden and NoteEdit notation programs. Programs to convert from at least six other formats exists.
Coding and readability	Plain-text Unicode (UTF-8) encoding. Human readable.
Stable format	Yes and no. Continuously developed, so LilyPond uses version number to specify which version is used.
Free of charge	Yes
Store all notation information	It stores normal notation. Extra features can be requested from the authors[35].

Table 6: Summary of LilyPond capabilities

3.3.5 MusicXML

MusicXML is a XML-based format for score representation. It is designed primarily to be an exchange format between different musical programs. The problem MusicXML sets out so solve is that most programs use their own file format, which makes it almost impossible to exchange written music effectively between users of different programs. A person using Finale may want to exchange a tune with a Sibelius user, who in turn may want to make some changes and send it back. This has been very troublesome until now since the two applications use different file formats.

MusicXML is gaining popularity as a good format for exchanging music. It is based on former work in music languages, namely Humdrum and MuseData. MusicXML is a XML edition of MuseData, extended with some concepts from Humdrum. It is text-based, and therefore completely platform independent. MusicXML is not made to replace existing formats, but it is made make possible the sharing of musical data. It can represent common western musical notation from the 17th century onwards including new and unusual concepts for 20th and 21st century music. Non-western music notation would use a different XML. Humdrum and MuseData was first of all made for classical and folk music. MusicXML extends this to also support concepts from contemporary popular music.

To make the file format interoperable between different types of music applications, the file format is based on the Extensible Markup Language (XML). XML has become accepted as a standard for the interchange of data between different applications as it can be used to describe data or metadata. The data is structured into hierarchical elements by using tags. XML can define a grammar for other XML documents in a special XML document called a XML schema. This means that XML is highly adaptable compared standardized markup languages like for instance HTML. The XML document can be validated that it conforms to the grammar in the XML schema. XML is intended to be readable by both computers and humans by having the XML file contain both data and metadata. A properly made XML file can be self explaining to a human by reading the tags. A drawback of XML is that the files can be large for even simple scores due to their verbosity. For example, the tag name is included both when the tag in turned on and when it is turned off. This makes XML less suited for manually writing scores directly in text editors.

For storing scores MusicXML has very much to offer. An example of the possibilities with MusicXML is that a single note can be defined to have a different duration when it is played by a music program than how it appears on the score. This is useful for instance for styles of music where the notes are interpreted differently than the written score. In jazz and traditional music this is not rare, as notes often are played with a swung feel that for simplicity is not written out in the score. By storing all information in the same file it can sound how it is supposed to sound while it still looks like people are used to. With the MusicXML 1.1 format also exact page formatting and playback information can be included. For example, there are tags that defines what instrument, midi-channel and midi-program a General MIDI compatible playback device (hardware or software) should use. This provides for easy conversion from MusicXML to MIDI.

A single note is defined in MusicXML by several tags. The following example shows some of the tags that can be associated with one note in a score. The code has been automatically generated by a musical notation program[18] and shows the coding of a singe C note in the 4th octave in MusicXML 1.1 format. Note that there can be fewer or more elements than in the example. From the example it is easy to see the self explanatory nature of XML and also how the tags build a hierarchy of informations. The pitch tag enclose information about the pitch as expected. The step is the note name, here it is C, and it is in the fourth octave. The step information is interpreted in the current key signature (defined elsewhere). A note also has a duration which is based on divisions of a quarter note. Here the duration is four quarter notes. The type sets the notated duration. Two definitions of duration may seem superfluous, but this makes it possible to have a different playback duration compared to the notated duration. A score can have several voices at the same time, and each voice can be set to a separate layer with the voice tag.

---

18 Finale 2004 using the Dolet 3 plugin

```
<note>
      <pitch>
            <step>C</step>
            <octave>4</octave>
      </pitch>
      <duration>4</duration>
      <voice>1</voice>
      <type>whole</type>
</note>
```

Example of MusicXML 1.1 coding of a 4/4 note of C in the 4th octave

MusicXML is currently supported by at least 55 different applications. Finale 2006 supports opening and saving MusicXML, and Sibelius 4 supports reading it. Through the use of plugins it is possible to exchange data between numerous music applications. For example, Recordare's Dolet plug-in for Finale (2000-2005) converts to and from MusicXML 1.1, and a similar Dolet plug-in enables Sibelius users to use MusicXML. The Dolet plug-ins also support batch-processing which saves a lot of work when handling big file collections. MusicXML is also supported by OMR-programs like Photoscore Professional, Smartscore and SharpEye Musical Reader. The Recordare web page [36] includes information on many programs supporting MusicXML. There is also software to convert between MusicXML and the GUIDO format.

The LilyPond package includes some programs to convert from other notation formats to LilyPond .ly-files. The program that converts from MusicXML to LilyPond has been tested briefly during the thesis work, but did not convert all information[19]. This makes LilyPond .ly files partly incompatible with MusicXML for the time being.

Parts of the MusicXML standard is has copyrights issues that makes it problematic to use with other major standards[37]. This may be a reason why it has not been a unconditional success with users and in the industry.

19 Simple scores were converted with acceptable results.

MusicXML	
Display score on screen	Yes, but batch mode direct conversion from MusicXML to image file software has not been found. MusicRain 2.0 is an interactive sheet music player that reads MusicXML 1.1 files and can be used to display MusicXML on screen on web pages. The format itself supports precise positioning of elements, so a script based display mechanism can be developed. MusicXML can be converted to LilyPond using the xml2ly program bundled with LilyPond, but preliminary testing suggests that the converter must be improved to get good results. If this is done, the LilyPond software can be used to display images of the score.
Print score on paper	Not directly. A program that generate PDF from MusicXML has not been found. Must be opened in a program that can generate PDF files. Here again, it can be accomplished through converting to LilyPond and use LilyPond software to make PDF. The MusicRain 2.0 Flash plugin can also be used to print MusicXML from web pages.
Hear the score	Yes. The format includes information for the playback not shown on normal sheet music which makes it possible to control the playback in greater detail. Can be converted to MIDI using other music software. However, script mode automatic conversion has not been found. Other options including a prototype java application called Xenoage MusicXML Player is available and MusicRain 2.0 can be used to listen to the score.
Hear and watch the position in the score simultaneously	Yes. MusicRain 2.0 provides interactive control in a web browser.
Let user user download and edit the sheet music	Yes. Can be edited in for instance Finale and Sibelius (may need conversion depending on version). The format is developed to be an Internet-friendly method to publish scores.
Include score in articles	No, not directly.
Software support	Support by over 55 applications. Supposed to be used as a file format to translate between other notation formats. Increasing support with music software.

MusicXML	
Coding and readability	The coding is Unicode text-based XML defined in DTD files defined by Recordare. The format can be written and read manually, but for simple notation other formats uses less code. Not easily readable by humans as the files may grow very big.
Stable format	Yes and no. Continuously developed, so MusicXML uses version number to specify which version is used. Current version is 1.1.
Free of charge	Yes
Store all notation information	It stores normal notation and layout information.

Table 7: Summary of MusicXML capabilities

3.3.6 Final Remarks on Notation Language

It is not easy to come to a final conclusion as to what is the best format for a music notation format. MIDI can easily be ruled out on the basis of its lack of graphical concepts. MIDI is not meant for storing notation, and too much information from the scores would simply be impossible to represent.

The other three represent attempts to make formats that should be adequate to express most musical concepts. The development of GUIDO seems to be discontinued without defining the Advanced and Extended version of the language. Because of this and the lack of software support, it is probably not the right choice for an archive or library of sheet music.

The last two, LilyPond and MusicXML, are both enjoying ongoing refinement and development. One advantage of LilyPond is that it supports direct editing in LaTEX, HTML and wiki documents. It is human read- and writable, and the language can include code to generate music using algorithms. Software for generating MIDI audio files and making images of the score for web and PDF files also exists. The output of LilyPond resembles the professional engravings done by hand in the past. Its support by major notation software such as Finale and Sibelius is however not good, which can be a problem for those not willing to input the scores without a graphical user interface. This can be remedied by the use of converter programs that accepts other formats and outputs LilyPond files.

The main advantages of MusicXML is its widespread support by an increasing number of programs for music and its built in support for defining how the music is supposed to be both played and displayed. It can also be easily verified by XML parser programs which exists in numbers. The problem with MusicXML is lack of current software to automatically generate image files, audio and PDF directly from the source MusicXML file[20]. Software to perform this functions can of course be developed, and might be under development already. MusicXML is supported by major notation software and optical music reading software.

20 The author of this thesis has not found such software

Both LilyPond and MusicXML seem to be mature enough to store the necessary information for the purpose of a sheet database. Which of the other two is most suited would probably require further testing. If better converter software between MusicXML and LilyPond is developed the choice becomes less important. The most obvious conclusion is that MusicXML seems to be able to store more information than LilyPond, and has more notation software support, where as LilyPond is easier to include in on-line documents and has tools for generating image, print and audio files.

4 Ingest of Sheet Music

The archive shall hold different kinds of digital media. Producing audio and video for the system is not described in this thesis as it is described extensively elsewhere. The reader may for instance consult the books “Designing Web Audio” by Beggs and Thede [38] and “Streaming Media” by G. Demetriades [39].

Digitizing sheet music is described here because sheet music will be a central part of the system. The choice of how sheet music is handled by the system also affects how the sheet music should be digitized and processed, and a common understanding of the possibilities is therefore in order. It is discussed here as this seems to be a less documented area than digitizing audio and video.

4.1 Digitizing Musical Notation

Digitizing sheet music from manuscripts can be accomplished in a variety of ways:

- Manually entering the scores using notation software with graphical user interface
- Manually typing the scores using a text-based score language
- Play the sheet music on a MIDI-enabled instrument and generate a MIDI-file and/or a score representation based on the MIDI-signals
- Scan the manuscripts and use Optical Music Recognition (OMR) software to generate a digital representation

What is the best choice will probably vary from case to case. Entering the scores using a text-based language like Lilypond requires some training to learn the language. If a person just wants to enter a single score, learning a new language might be a lot to ask. However, writing simple scores is supposed to be quite easy with LilyPond, and no expensive software must be purchased.

Entering scores using software with a graphical user interface also requires some getting used to, but people working with sheet music regularly may already be trained in using such software. Advantages of a graphical user interface is that the user instantly sees how the score will appear when editing the score. Examples of software are commercial programs such as Finale, Sibelius and Harmony Assistant[21]. There are also some free programs available; examples are GNU Denemo[22](for Linux) and iabc[23]. These programs let the user place notes, other musical symbols, lyrics and text directly in a score on the computer screen. After the score is finished, it can normally be printed on paper (or to a PDF-file) or saved to file. Most programs have their own native file format, but some support exporting to file formats that can be opened by other notation software from other vendors[24].

21 Harmony Assistant web-page: http://www.myriad-online.com/en/products/harmony.htm
22 Denemo web-page: http://denemo.sourceforge.net/index.html
23 iabc web-page: http://abc.sourceforge.net/iabc/

Several of the programs also support entering music by using a MIDI-enabled device like a MIDI-keyboard. This alternative often requires some adjustment to the finished score, since there may be several ways for the program to interpret what is played. It is practically the same as importing a MIDI-file into notation software, and the same errors may appear[25].

The last of the mentioned alternatives is to scan the manuscripts and let computer software automatically convert them to digital sheet music. This option has been given some extra consideration in this thesis work and is dealt with in the following chapter.

4.2 Optical Music Reading

A lot of work can be saved if a computer can do the work of interpreting sheet music manuscripts into a digital representation instead of having a person to manually do the job. Software to optically read normal text has been around for years, and is in great use. The technology is called Optical Character Recognition (OCR). The technology has matured into being able to interpret machine typed text quite well, also from schemes and forms.

Sheet music is a bit harder to interpret than written text. This is due to several issues regarding how sheet music is engraved. Sheet music has a two-dimensional structure with complicated symbols. The symbols vary in size and length, and the placement of the symbols in relation to the notes can also vary very much. In addition there are physical aspects with the paper as varying backgrounds, the sheet may be folded and small spots may be interpreted as a symbol. Up until recently software for scanning sheet music has not been performing very well. The software has been slow, the error rate has been high and there have been problems with interpreting complicated symbols. A OCR-program may use dictionary functions to spot errors, a method that is not directly transferable to OMR-software. Some errors like the wrong number of beats in a measure may be spotted/corrected though. The wrong pitch may be easier to guess for a human than computer software. One reason that OMR-software lags behind OCR-software may also be the simple fact that the number of users of OCR is far bigger than for OMR. A developer of OCR-software will therefore have more potential costumers and a bigger economic potential.

There are several software systems for Optical Music Recognition available. Listed are some of the available programs:

Commercial software:

- PhotoScore[26]
- SmartScore[27]
- SharpEye Music Reader[28]
- Vivaldi Scan[29]

24 Sometimes saving to other formats is done using plug-ins. An example is the Dolet plug-in for Finale which saves and opens the MusicXML-format.
25 see the MIDI-section (section 3.3.1)
26 Photoscore web-page: http://www.neuratron.com/photoscore.htm
27 SmartScore web-page: http://www.musitek.com/
28 SharpEye web-page: http://www.visiv.co.uk/
29 VivaldiScan web-page: http://www.vivaldistudio.com/Eng/VivaldiScan.asp

Shareware software:
- OMeR[30] (Optical Music easy Reader)

Open-source software:
- Gamera

The scanning software mostly requires some manual adjustment before the file can be saved. Some programs can also be integrated with notation software, and are included in light versions with the notation software releases.

The PhotoScore software has been tested as a part of this thesis to get an idea of the level of success of modern OMR-software. This test can be found in section 4.2.1. The Gamera system represents another approach to sheet scanning. It is based on training the software to understand different types of documents, and can be trained to understand handwritten notation. This differs from PhotoScore and most other commercial product which can not be adjusted to such extent. The Gamera system for sheet music recognition is described in section 4.2.2.

4.2.1 Photoscore

PhotoScore is a commercial program designed to scan and read sheet music from files and paper into digital sheet music. It is not designed to read hand-written music, only printed music. It can export in several formats, including the MusicXML-format, MIDI, NIFF and WAV. It can also send the score directly to Sibelius from the file-menu.

PhotoScore is equipped with its own interface for editing and correcting the scanned sheet music. The User Manual suggest at least making sure that the scores have the correct timing before exporting to other programs[40]. This is to make sure that no information is removed due to wrong time signatures. If the bars contain to many beats information may be lost when converting between different programs. The program interface for editing the score closely resembles Sibelius, and uses a combination of mouse drag-and-drop and keyboard shot-cuts.

4.2.1.1 Test

PhotoScore 4 Professional has been tested to get an idea of the current state of commercial OMR-technology and PhotoScore has been chosen as a representative of this technology.

The testing was done with a HP PSC 1215 flatbed scanner. The PhotoScore software has been tested on a standard Windows XP-based laptop running on a Intel® Pentium® 1400Mhz processor with 768Mb RAM.

The user interface of PhotoScore is quite intuitive and easy to get used to. After a score has been read and interpreted, the user can compare the result with the original by moving the mouse over the score. The original scanned image is then showed above the score and the user can compare the result note by note without looking at the original paper. Editing the scores for errors is done by dragging and dropping notes and symbols with the mouse. The user

30 OMeR web-page: http://www.myriad-online.com/en/products/omer.htm

interface is almost the same as that of the Sibelius notation software, so Sibelius users will probably find it familiar.

Six different scores were tested. They ranged from simple sheets with one voice containing only basic notation to sheets music with multiple staff systems and more advanced notation including slurs, dynamics, rests, hairpins and articulations.

The testing shows that even simple sheet music often contain a certain amount of errors, and that in these cases, little time is saved compared to using music notation software from scratch. The error rate of the tests varied quite a bit. Some standard printed music was read quite successfully while other sheets seemingly equal in quality produce lots of mistakes. The tests also confirmed that even neatly handwritten sheets produced bad results even though they are easily read by the naked eye.

Some problems seemed to repeat themselves:

- Chord names (A, Em, Bb, etc..) written below the staff they belong to are wrongly connected to the staff below. There does not seem to be a quick way to fix this.
- Repeat bar lines were mostly interpreted as single bar lines, and had to be edited.

More testing should be done before it is possible to make substantial conclusions. For printed music time can be saved compared to input the score using notation software, and one can expect the OMR-technology to further improve in the future.

For handwritten manuscripts other software such as the Gamera system should be considered.

4.2.2 Gamera

Gamera is a document recognition framework that can be adjusted/programmed to recognize many different types of documents. Originally it was made to do score recognition, but it has evolved into a general document recognition framework especially suited for large scale digitization projects. In addition to sheet music it has been used to make recognition systems for areas as medieval manuscripts, Navajo language documents, eighteenth-century census data and lute tablature[41].

The system can be used by domain experts to build custom systems. It uses a combination of programming libraries and GUI-tools to develop recognition systems interactively. The goal of the development of Gamera was to “support an efficient test-and-refine development cycle”[42]. The software is open source and can be run on many different operating systems (including Windows and Mac). It can do batch processing, which makes it capable of processing large amount of documents without user intervention. It can also give a measure of the success (level of confidence) so that the users can take other measures if the recognition fails.

The Gamera system supports the following tasks, which may be arbitrary complex or removed completely depending on the problem[43].

- Pre-processing
- Document segmentation and analysis
- Symbol segmentation and classification
- Syntactical or structural analysis
- Output

The next two figures shows a typical working situation with Gamera (image sources: [44]). Figure 7 shows the scanned sheet in the upper left corner of the screen. Under, in the lower left corner, the same picture is shown after some basic image processing. The user console is in the upper right corner, where commands may be entered. These commands may also be scripted. The lower right corner shows an image histogram. What windows are shown during a user session can vary substantially.

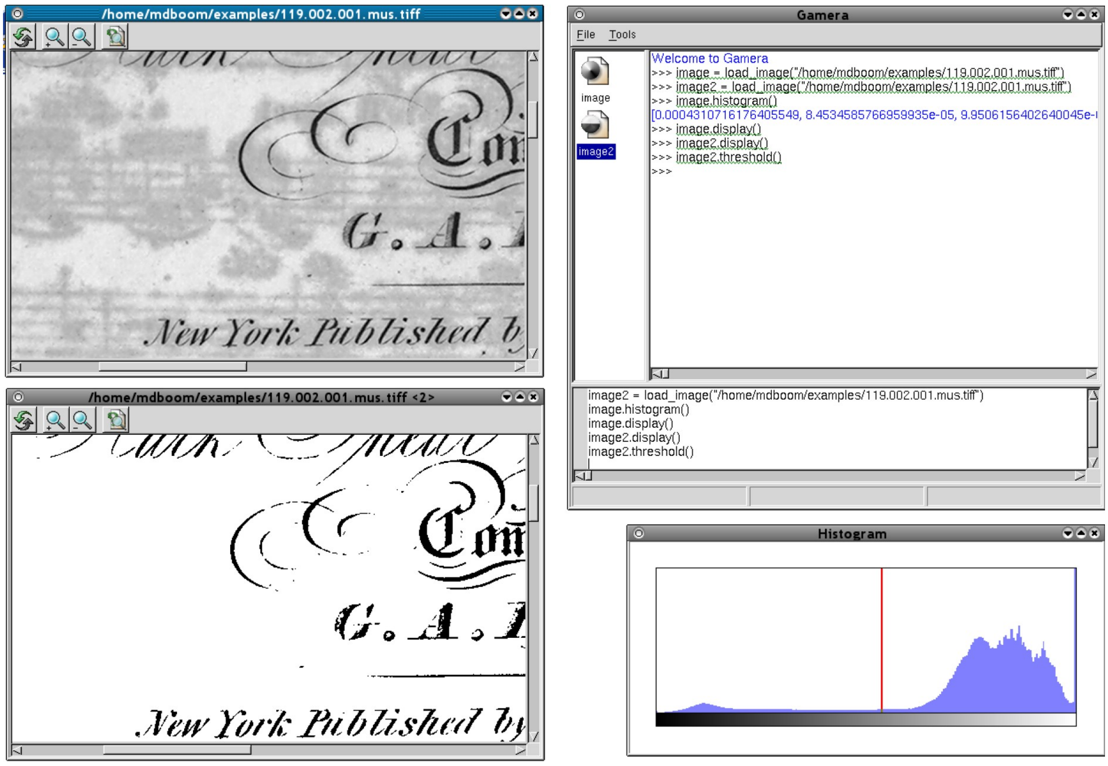

Figure 7: A Gamera session showing the image viewer, image analyzing tools and the command console.

Figure 8 shows a training session. Gamera has extracted symbols from the scanned sheet, and the user classifies the symbols into groups of similar symbols. When the next document is processed by Gamera, it uses the classification to make guesses for which groups each symbol belongs to. Any errors can be corrected by the user, and after a few “training documents” the system will be able to classify the documents with fewer and fewer errors. When for instance one sheet collection is to be digitized, the collection may be quite uniform if it is engraved by one person. Then the first documents are used for training, and the rest can be processed automatically, saving a lot of work for big collections.

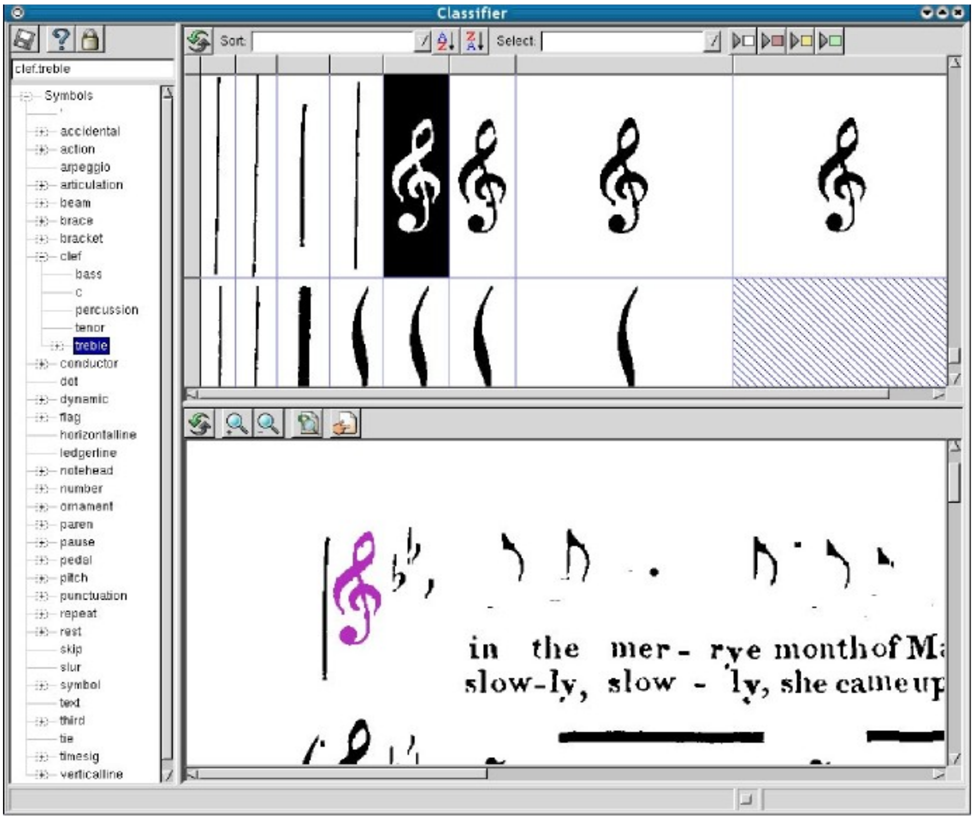

Figure 8: A Gamera training session is shown where the user is classifying symbols.

4.2.2.1 Scanning of Musical Scores

Recommendations for storage, web-delivery and printing of musical scores can be found i [45]. The article also have recommendations for scanning of musical scores. Testing shows that three pixels per detail is needed to obtain good results using OMR-software. The smallest detail should be be measured before the scanning process, and then this info should be used to set the scanning resolution. However, the details of sheet music is often smaller than 1 mm. Measuring such small details requires special instruments. The general recommendation is therefore that 600dpi resolution is enough to catch the necessary details of written notation. 600dpi means that details as small as 0.027mm is scanned with 3 pixels. Using a higher resolution does generally not give better results, even though it is suggested that tests are performed with the particular scanning and OMR-software solution [45].

5 Other Archives

To get an idea of technologies in use today some existing on-line music databases for sheet music, both payment based and free, has been examined. Video sharing sites have also been looked into to get a brief understanding of its workings. Finally two planned digitizing projects are described, as they represent some of the many ongoing digitizing projects around.

5.1 On-line Systems for the Sale of Sheet Music

To get an idea of how sale of sheet music can be done via the Internet, one representative on-line sheet music database has been reviewed.

5.1.1 SheetMusicNow.com

Url: http://www.sheetmusicnow.com/

SheetMusicNow.com is a sheet music database containing over 15.000 scores. Genres includes jazz and classical music.

To download, print and view the scores, special software has to be installed on the users machine. The software, called SafePublish, handles DRM by making contact with a server each time a user wishes to print scores. The printing works as follows:

- The user download and installs the SafePublish software (free, 1.4MB).
- The score(s) the user has bought and downloaded is opened in SafePublish.
- When the user prints from SafePublish, the program checks with a server in Copenhagen if the file has already been printed. This check makes sure that the score can only be printed once, which is what the user pays for.

The way SafePublish works, it is not possible to use the scores in the users own music software. This is intended because the site sells copyrighted material and they want to avoid that users spread the material.

The following terms can be searched: Title/Catalogue number, Composer/Arranger/Editor, Publisher and Instruments. There are also different categories that the user may choose from like instrument families, instruments, styles and artists.

It is not possible to perform content-based searches[31]. After the search has been made, the user is presented with a description of the results. The results includes the following information: The title, composer, ensemble/instrument information, musical style, arranger, editor, publisher and the price. SheetMusicNow do not have sound preview of the sheet music. After the search, it is possible to preview the first page of the score, or add it to the shopping basket.

31 Content based search: see chapter 9.2.2 “Content Based Query”

Figure 9 shows the search results after a search for "canon" by "pachelbel" was performed. The four titles found are listed, including information about the title, key, ensemble type, style of music, arranger, editor, publisher and the price.

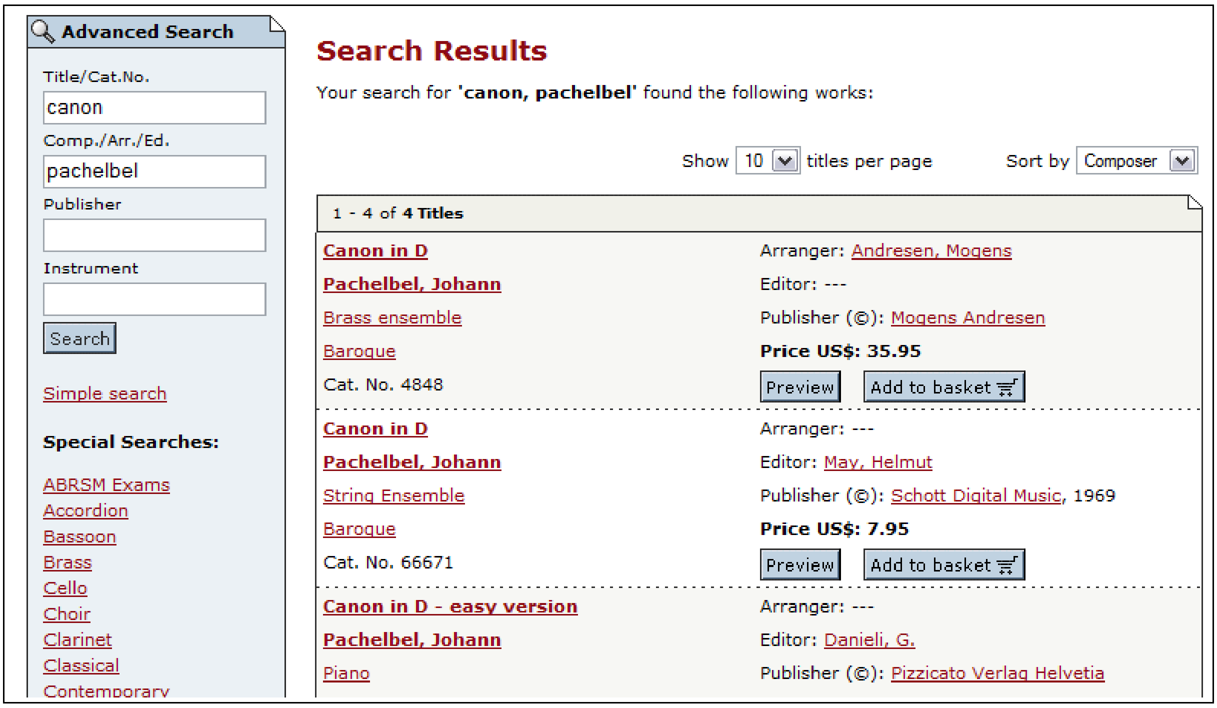

Figure 9: A search result from SteetMusicNow.com includes information on title, composer, ensemble, style, price etc

5.2 *Free On-line Music Databases*

There are numerous on-line music archives on the web today. They span from simple web-sites containing MIDI-files for downloading, to advanced sites offering musical scores, audio recordings and any kind of related information. Some are completely free of charge, and some sites charge either per download, a subscription fee, or a combination of these. To better understand the possibilities of a musical archive and to investigate the possibilities for reuse some current on-line music projects have been evaluated.

5.2.1 Choral Public Domain Library (CPDL)

Url: http://www.cpdl.org/

A database with over 7000 scores by over 1000 composers. Started in 1998, it claims to be one of the largest free sites for downloading musical scores. Most scores can be viewed (and downloaded) in .PDF-format and listened to in MIDI-format. Some scores are also provided in Finale or Sibelius-format. The web-design is based on Wiki, and the main page provides searching possibilities, and links including to listings of composers, scores, how to contribute to the project and upload sheet music.

The site also has forums with different topics; administrative discussions, support, music requests, member meeting places and so on. The forum is based on phpBB, an open-source PHP-based forum.

Like the Wikipedia project, CPDL is based on people around the world contributing with information and scores. Only work which holds no copyright may be uploaded. The composers listing only holds information on composers that have been dead at least 70 years or CPDL contributors.

CPDL does not provide for sound other than MIDI interpretations of the scores. CPDL has many good solutions for sharing scores, and added the ability to share sound and video recordings along with a standardized format for scores CPDL could be a good starting point.

Pros:

- Downloadable score music
- Forums for discussions, questions etc.
- Good model for a collaboration on the net
- Well arranged design based on the Wiki-project which is open-source and free to copy
- Open for contributors
- Everybody control everybody, in essence an open control function
- PDF and/or PS files for printing sheet music

Cons:

- No support for sound and video (currently), except simple MIDI-playback of scores.
- No standard file format for scores. Formats used are Finale and Sibelius, where for instance the different Finale versions use incompatible formats of .mus-files.
- Only intended for choral music

5.2.2 The Mutopia Project

Url: http://www.mutopiaproject.org/

The Mutopia Project is a web page where users can view, download and print scores for free. It is based on the idea that volunteers contribute to the collection by typesetting sheet music editions with GNU LilyPond. Originally meant for old material where the copyright has expired, it is also used by people to offer new music, arrangements and editions to the world for free, and with few restrictions. Searching is based on metadata. The user can search free text or in fields that narrow the search down to composer, instrument, style, LilyPond version and for when it was added or edited. The collection can also be browsed by composer, instrument and style.

People who want to contribute to the Mutopia Project may contribute directly from the Mutopia Contribute web page[32] without registering. Fields for title, composer, instruments, style etc. must is entered into a form, and sheet notation in LilyPond language is uploaded. The Mutopia Projects reserves the right not to publish music which they see unfit for the site.

The file formats offered for download are:

- Postscript (.ps), in A4 and letter sizes
- PDF (.pdf), in A4 and letter sizes
- LilyPond (.ly)
- MIDI (.mid) for previews

They are also planning support for the MusicXML for input, but are waiting for a better program to convert from MusicXML to LilyPond than the current program xml2ly, which seems to be out of development[33].

As a conclusion on the Mutopia Project, it seems well suited as a web site for distributing sheet music that has no copyrights. Old and new work can be added by all users, as long as the work has no copyright issues. Only metadata information can be searched, and the meta information is limited. The is not content based search. The system has MIDI audio preview, but otherwise it has no support for audio or video.

5.2.3 Musipedia.org

Url: http://www.musipedia.org/

Musipedia is a database of sheet music which is free for everybody use. It is based on open-source systems like GNU LilyPond for scores, TYPO3 for content management and MySQL for databases.

To edit or add music, it is necessary to register with name and e-mail address. After registering, access to different discussion forums is granted in addition to being allowed to edit scores already in the database. The topics are different aspects of Musipedia. When adding a score, the following fields must be filled in:

- Which license to apply for the material
- Composer or Band/Group (depending on category of music)
- Title
- Category (Classical, folk music, popular, hymns and carols, national anthems)
- Lyrics and other textual information
- Parsons code for the music

32 http://www.mutopiaproject.org/contribute.html
33 xml2ly by Guido Amoruso (shipped with GNU LilyPond)

- A Midi file and a picture file is uploaded OR LilyPond / ABC[34] source code is entered. LilyPond source code is preferred, and if that is provided, image files and MIDI files will be generated automatically by the server.

Musipedia has some advanced search options available. As of May 11, 2006, the Musipedia contains 30834 pieces of music, and has indexed 15751 MIDI files from the web. In addition to text search based on metadata associated with the score, the Musipedia database and/or the Internet MIDI files can be searched with melody search. With melody search, a segment of a known melody is input to the search algorithm and compared to the database. A melody can be input the following ways:

- Query by humming (A java interface where the user can hum or sing a melody or upload a wav-file)
- Melodic Contour search (The contour indicate the pitch change of to consecutive notes, for instance: up-down-down-same-same-up...)
- Rhythm search. The user can tap the rhythm using the keyboard.
- Piano Screen Keyboard search. The user can input notes by clicking with the mouse on a keyboard image with different durations. The image of the score below the keyboard is immediately updated. Piano Screen Keyboard is shown in Figure 10.

34 ABC is a notation language typesetting sheet music.

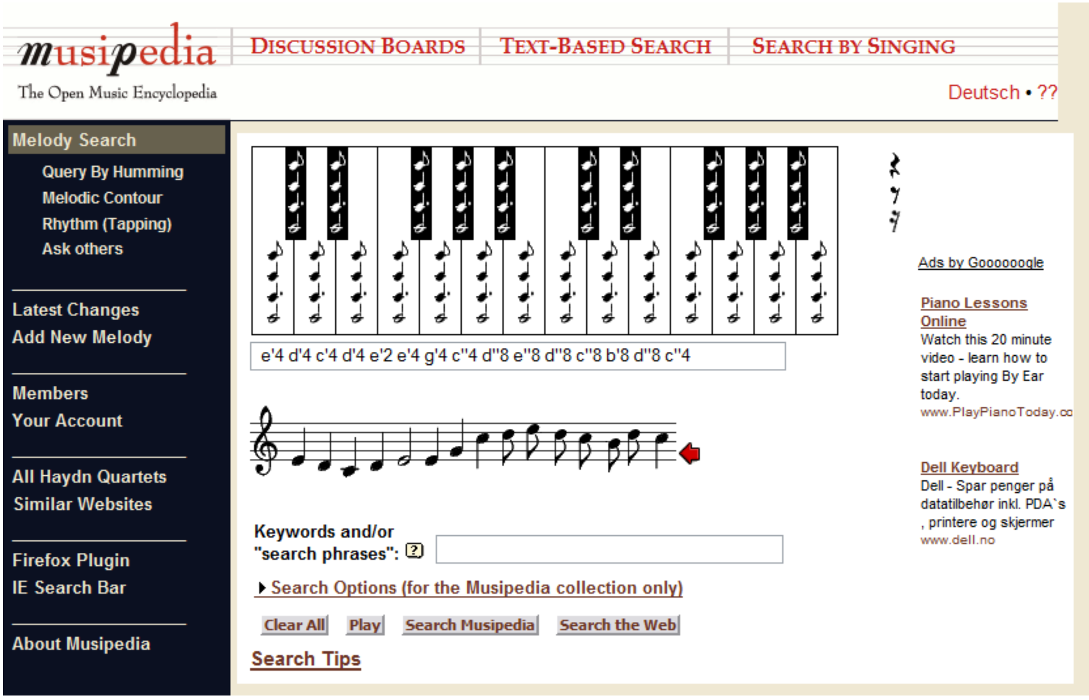

Figure 10: Melody Search with Musipedia. The figure shows a piano where the user can click to input the melody element to search for, and the image of the score in the middle of the figure is automatically updated.

Pros:

- Easy to search for and download sheet music
- Users can edit works with mistakes. The original contributor can be notified automatically of this to prevent users from hampering with the work.
- Midi preview of sheet music.
- Collaboration for everybody

Cons:

- No support for: payment, sound, video.

5.3 On-line Digital Archives for Video

Two different system that allow users to view video content on the Internet are examined in this chapter. The first, Filmarkivet.no, is an example of a payment-based download site[35]. The focus here is the payment options and the digital rights managements. Full details on the DRM-technology has not been found so the description will be somewhat simple. The second type are websites which allows users to share videos.

35 Some videos are offered for free, but most have a price.

5.3.1 Payment site: Filmarkivet.no

Url: http://www.filmarkivet.no/

Filmarkivet.no is a Norwegian digital film archive available for private, public and commercial users. It was opened to the public in 2005, and is for the time being restricted to users in Norway (based on IP-address). The service is accessible via PC and some television cable networks in Norway.

The archive offers movies concentrating on the history of Norwegian Film. All categories of movies are represented, feature film, commercials, cartoons and so on. Also a selection of foreign movies is available. Movies from the National Film Archive (det nasjonale filmarkivet) at Norsk filminstitutt are restored, digitized and made available after gathering the right permissions[46]. Figure 11 shows the web interface of Filmarkivet.no after an old Norwegian commercial called “På vippen” has been selected. The figure also shows how the films are categorized in different genres, countries and categories.

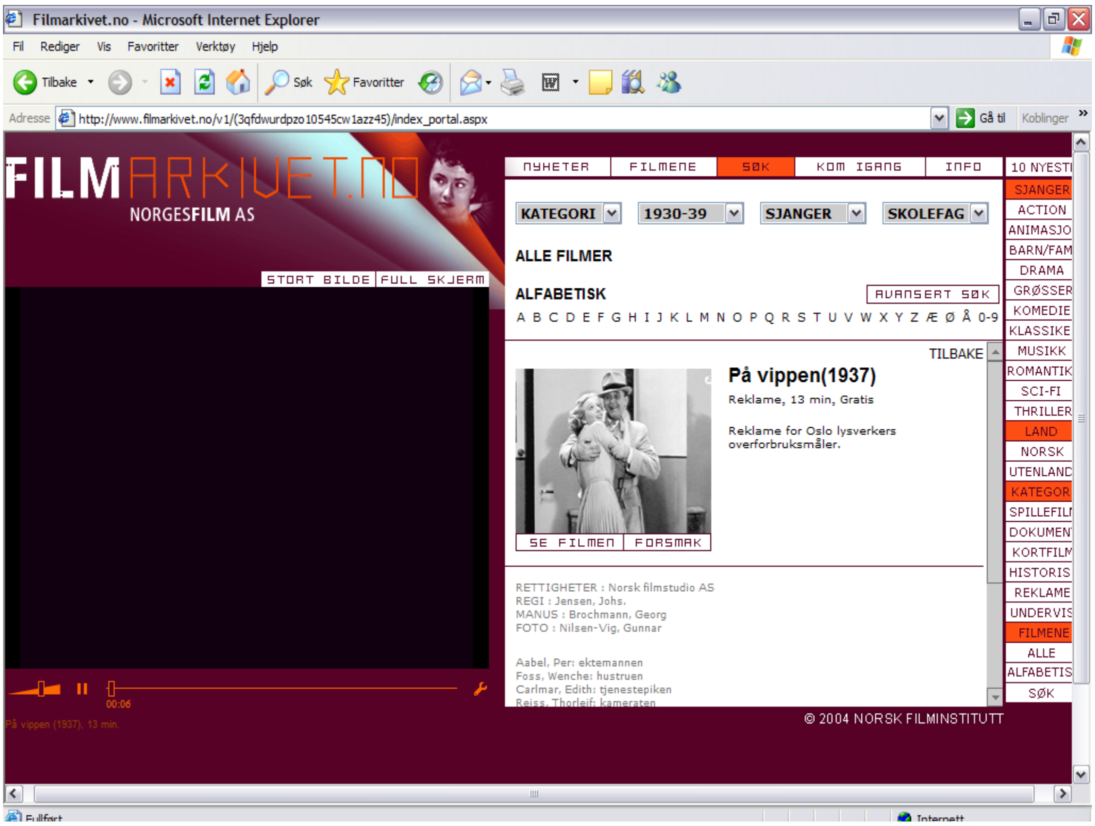

Figure 11: Screenshot from Filmarkivet.no after the commercial “På vippen” has been selected.

Filmarkivet.no is made by Norsk filminstitutt in co-operation with Agder Energi Bredbånd AS and NorgesFilm AS. The PC and IP-TV service is developed by Norsk filminstitutt and Agder Energi Bredbånd AS. Filmarkivet.no is operated by NorgesFilm AS.

Payment solution:

- Telenor and NextGenTel customers may pay for downloading movies on their regular invoice.
- Payex[36] (for other customers)

The technologies used are primarily from Microsoft. This restricts the user base to users with Windows Media Player compatible systems. This may be a problematic design solution if the users have different operating system and web-browsers, for instance the service does not work for Mac users or other browsers than Microsoft Internet Explorer. The films are secured with DRM components from Microsoft. To use the archive, the user must agree to use the latest DRM components of Windows Media Video. This includes anonymously registering to a server run by Microsoft, which is contacted when a movie is played[46].

5.3.2 Video Sharing Sites

A system for collaboration could also offer the possibility of sharing videos between users. Several existing web sites are offering users to share their personal videos. Some sites are web-based, which means that the user access the site and upload videos using the web browser. Other require installation of special software to upload videos. They normally allow uploading of a wide range of video file types. Some of the sites convert the movies to one type of movie like Flash[37] or QuickTime. Some also allow uploading from mobile phone or PDA.

Examples of such web-client based sites include[38]:

- YouTube
  http://www.youtube.com/
- CastPost
  http://www.castpost.com/
- ClipShack
  http://www.clipshack.com/
- DailyMotion
  http://www.dailymotion.com/
- Grouper
  http://www.grouper.com/
- OurMedia
  http://www.ourmedia.org/

36 See 7.1.2
37 Macromedia Flash is a method of adding animation and interactivity to web pages. It supports vector and raster graphics, a scripting language and streaming of audio and video.
38 All listed web sites were checked June 1, 2006.

- Revver
  http://www.revver.com/
- Vimeo
  http://www.vimeo.com/
- Social
  http://vsocial.com/

5.4 Other Planned Digital Archives

During the work with this thesis I have come across two other major digitizing projects in Norway. These two projects are briefly mentioned here as experience from these projects could be of use also for this and other digitizing projects.

5.4.1 NRK

NRK[39] has one of the largest archives of published records in Norway. These recordings are used in the production of radio and TV-shows. Recently (march 2005) NRK has landed on NSA, NorCom and CognIT to develop a solution for a digital music archive. The system will be a digital archive that holds all the commercial records owned by NRK making them readily available for all production lines. It is part of NRK's plan of establishing a 100% digital production environment for their radio and TV-productions[47]. The project is the first of it's kind in Europe, and other broadcasting companies are paying attention to it's development. NRK also have plans to digitize the rest of the production chain, which includes migrating all their stored audio and video to a search friendly digital archive. The archive is supposed to be accessible for NRK journalists from both the main office and the district offices and will be integrated with the production of new shows and programs[48].

5.4.2 MemNor AS and Norsk Lydinstitutt

Norsk Lydinstitutt (Norwegian Institute of Recorded Sound) in Stavanger is an archive containing commercial and private recordings. It is one of the largest private collections in Europe with more than 100,000 LP's, 40,000 78 rpm recordings plus audio reel tapes, cassettes, videos and books on music. Plans for the archive include digitizing the collection to preserve it for the future and to be able to serve the public in a more efficient way[49].

Norsk Lydinstitutt and the Belgian company Memnon Audio Archiving Services SA are establishing a company called MemNor AS that will offer sound archiving services. They plan to cover the whole chain of digitizing including: indexing, digitizing, restoration, production of copies, long-term storage of both digital and analogue material, migration, digital asset management and profiling of the collection. The services will be offered to sound archives and institutions with sound collections. They plan to follow international standards for long-term storage of the audio heritage, and will be able to tailor their offerings according to the customers needs and resources[50].

39 Norsk Rikskringkasting – The Norwegian Broadcasting Corporation

6 Existing Software

Much resources and time may be saved by using existing software to develop the system. Depending on the nature of the software and the availability of source code, integrating existing software systems can be done directly or by developing new code to glue the system together. A promising digital library software system is described in the next section followed by an example of a content management system.

6.1 Greenstone Digital Library Software

There already exists software that can be used to implement a digital library. One example is Greenstone which has been used for digital libraries with content such as audio, video, texts, books, pictures and much more. It has been used to build the BBC radio and TV archive[51], in addition to many other public and private digital archives.

Greenstone is made by the New Zealand Digital Library Project at the University of Waikato in New Zealand. It is developed and distributed in cooperation with UNESCO and the Human Info NGO[52].

Greenstone is a framework for building digital library systems. It aims in particular at providing a platform for libraries, universities and public institutions. It is developed as an open, flexible architecture. Some of the key features are listed below[2][53]:

- Open-source software (released under the terms of the GNU General Public License)
- Runs on Unix and Windows
- Accessible via Web browser
- Multilingual – End user interface in over 30 languages
- Handle documents in any language
- Standard, open document model that can be readily exported into other digital archives and libraries
- Plugins are used to support ingesting documents and extracting data from files.
- Supports a variety of documents, support for more document types can be achieved by making a new plugin
- Supports advanced documents. For instance; a picture may be a document, but a HTML-page with pictures may also be a document. It separates between files and documents
- Full text indexing of documents
- Automatically creates access structures
- Handles collections of text, audio, pictures and video
- Permit authentication of the users
- Provides user logging
- Administrative function which gives specified users the right to authorize other users to build collections, view collections and so on
- Supports distributed collections

- Support multi-gigabyte collections
- Dynamically adds and updates new collections
- Offers full-text and fielded search

Development of a new version, Greenstone 3, is ongoing. The new version is supposed to be more dynamic, configurable and extensible. It is based on a network of independent modules that communicate with XML messages. Addition of new modules is simplified. Some highlights[51]:

- Retaining the advantages of Greenstone 2: Multiplatform, multilingual and highly configurable
- Modular based topology that can be spread across many servers
- Modules can be loaded at runtime
- Open-source code (GPL)
- METS framework is used to represent documents internally – both content and metadata. METS is a open framework that support interoperation with other library systems and avoid duplicating data
- Built using Java code – provides platform independence
- Dynamic loading of objects
- XML-messages between modules. Messages can be streamed across a network using SOAP. All modules can describe themselves in machine readable form.
- XSLT can be applied to messages to transform them to other forms, for instance human readable forms for presenting information to humans or sort query results
- Dynamic. Collections and modules can be changed while the system is running
- No centralized manager process

Judging by the example digital library of archive systems that are described in various document about the Greenstone system, it seems to be a system that will be able to support many of the requirements of our system. Since the development of Greenstone 3 is not finished it is not recommended for users yet. Greenstone 2 is recommended[52].

6.2 Content Management Systems

Content management systems is software that allow users to collaborate on different tasks like creating documents or articles. Content management systems can require the user to use special software for collaboration, while other systems can be used through the web browser alone. A list of systems can be found in [54], and a more thorough explanation of the different types of content management systems can be found in [55].

6.2.1 MediaWiki

MediaWiki is a free content management system used by many web sites. The most well-known MediaWiki system is probably Wikipedia®[40], a multi-language free encyclopedia.

40 See http://www.wikipedia.org/

MediaWiki is a free server-based software. It is licensed under the GNU General Public License (GPL). The software is designed to be run on a large server farm for a website that gets millions of hits per day. MediaWiki is an extremely powerful, scaleable software and a feature-rich wiki implementation, that uses PHP to process and display data stored in its MySQL database. MediaWiki is designed to handle a large number of users and pages without imposing too rigid a structure or work flow.

Requirements (server side):

- Any operating system that run the following components (GNU/LINUX suggested)
- Any web server (Apache2 suggested)
- Database system: MySQL
- Scripting: PHP

There are multiple sites containing music which use MediaWiki:

- ChoralWiki (http://www.cpdl.org/)
  - Described in 5.2.1
- The Folktunes Archive (http://folktunes.org/)
  - Instructional, vintage, and demo recordings, lyrics, and all thing having to do with folk music, editable by all.
- Partitures.net (http://www.partitures.net)
  - A Catalan and Spanish Wiki for free music scores and music education material.
- Minnewiki (http://minnewiki.org/)
  - A Minnesota Music Database run by Minnesota Public Radio. Articles about Minnesota music; people, groups and venues.
- Tous aux Balkans! (http://tousauxbalkans.free.fr/)
  - Music, scores, lyrics, translation, dance steps. Music from the Balkans (Greece, Romania, Bulgaria, Serbia, Gypsies).
- KiwiMusic.Info (http://kiwimusic.info/)
  - A wiki database for New Zealand music.

6.2.1.1 WikiTeX

WikiTeX is an extension to MediaWiki that makes it possible to edit scores in the LilyPond format directly in Wiki articles. The music notation is written in the document with tags to separate the LilyPond notation from the rest of the text, and the WikiTeX software automatically replace the music notation with an image of the score. For articles where several users collaborate the scores can easily be edited with a edit link or listened to via a link to a automatically generated MIDI file[56].

6.2.2 Discussion Forum

Discussion forums is a popular feature of many on-line collaboration sites. The normal way such forums work on the Internet is that they are structured in threads with different topics. Users may start threads and answer other users threads. The postings are normally visible to all other users, and the discussion take place without all being on-line at the same time. Users can not alter other users messages. Content management systems often have built in discussion forums, but many stand-alone products also exists.

6.2.2.1 PhpBB

One example is phpBB which is a fully scalable and customizable open source bulletin board package. It is free to use and modify, it has got a user friendly interface and is supported by a big community of users and developers. It runs on several common database systems including MySQL 3.2x, PostgreSQL 7.x, Microsoft SQL Server 7/2000, Microsoft Access (via ODBC) with more to come. It is designed with security in mind, with powerful authorization systems, and very flexible forum design and maintenance functions[41].

41 See http://www.phpbb.com/features.php for a full list of features.

7 Payment Systems

The system may be supposed to handle payment for services. The system could for instance handle charging for downloading a research report or a music file.

There are an abundance of ways to achieve this, and exploring all possibilities in detail is not intended in this chapter. Some available solutions are briefly summarized, but it should be kept in mind that there may be need for some advanced charging mechanism. Some users may for instance have free access to services which other users shall be charged to use, and this should be supported by the payment system.

Developing a new payment system from scratch may be quite expensive. Instead of using money and resources to develop a new system it should be considered if an existing system can be used. The KOPEK Payment system seems like a viable alternative, and chapter 7.1.1 gives a summary of KOPEK features. A few other options are mentioned in chapter 7.1.2.

7.1.1 KOPEK Payment Server

One payment solution is the KOPEK Payment Server. It is a complete solution for the sale of all kinds of content, services or physical products. It requires no programming or changes to the existing web site, and can be seamlessly integrated with existing systems[57][58].

The system includes [59]:

- Access control with users and user groups
- Different pricing for different user groups
- Wide range of payment methods including:
  - Pay per object, per access, per subscription, time limited etc.
  - Charge credit card, mobile phone or bank account
  - Charge for any action/function/item
- Security
  - PKI-based
  - Certificates
  - Digital signatures
- Privacy
  - Data encryption
  - No necessary link between id and real life identity
- Cross platform support
  - Windows CE/95/98/ME/XP/2000/NT 4.0, Linux, FreeBSD, Solaris
- Software development kit (SDK) for Java and C++ languages
- User identification, authentication, authorization and tracking
- Multiple language/currency support

- Support refunds
- True single sign in

All in all this seems to be a solution that can be integrated with most web solutions and payment variations. KOPEK support differentiated pricing for different users and user groups. The software is free to install and use, and can be installed on a server that sits between the original central server and the Internet as shown in figure 12. The KOPEK Server software may also be installed on the existing web server.

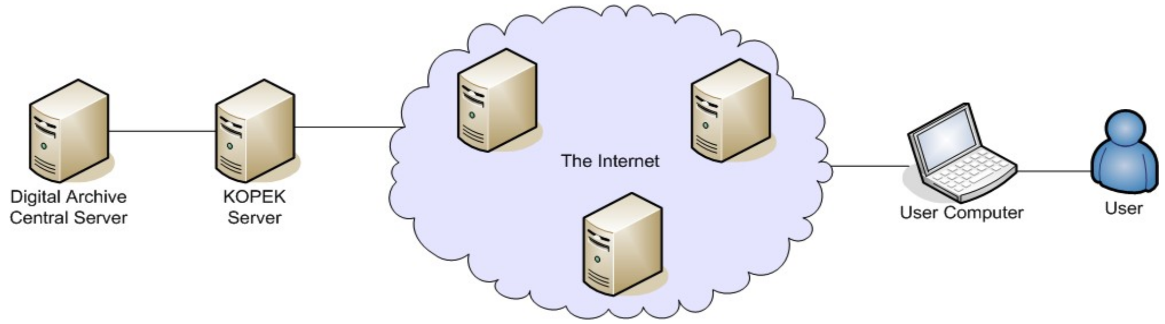

Figure 12: The KOPEK Server is shown installed between the existing central server and the Internet. The KOPEK Server can then handle payment for objects in the archive.

The cost of use is per March 2006 based on transactions fees (4% of transaction value + £ 0.002), withdrawal fee (£3.00 per £100.00) and fees for transferring money to banks [59].

7.1.2 Other Payment Options

Other ready-to-use payment options are (not a comprehensive list):

- PayPal – http://www.paypal.com/
  - Several solutions to let merchants accept payment over the Internet.
- Payex.no – http://www.payex.no/
  - Norwegian payment solution that functions like an electronic wallet. Used by over 250 existing Norwegian websites. Support credit card, user accounts, SMS-payment and more [60].
- Authorize.net – http://www.authorize.net/
  - Manages credit card or electronic check transactions for merchant sites.

8 Rights Management

Digital Rights Management (DRM) concerns handling the rights to use, watch, edit, buy and sell material with intellectual property rights. DRM technologies are technology to enforce such rights, for instance copy protection schemes.

For many types of digital media the publisher may want to restrict how the media can be used. A folk music archive may for instance want to restrict the possibility that users make their own copies of certain recordings and record companies do not want people to share perfect digital copies of commercial recordings for free.

DRM technology can make it very cumbersome to copy digital data, but if for instance a sound recording is to be played through the users sound card it is impossible to guard against making a recording of the sound signal. The user may simply connect sound recording equipment to the sound output of the sound card. Although the copy is analogue, subsequent copies may be made digitally and hence perfect. The file itself may be encrypted, but it has to decrypted sometime to be playable. If the decryption information is hidden in the playback software it is likely that some people see that as a challenge that must be solved. The next example shows some of the problematic nature of DRM:

iTunes is media player software for PC and Mac. It can be used to connect to the iPod portable media player and to the iTunes Music Store where music can be purchased and downloaded. iTunes only support two operating systems, Windows (2000 or XP) and Mac (OS X v10.2.8 or later)[61] and use various techniques to support DRM. For instance iTunes use protected AAC[42] encoded audio files. According to [62] they also only allow songs to be played at three different computers and only support Apple iPod as playback device. Many users feels this as an infringement of their rights. Linux users and users with other mp3 players than iPod are for instance prohibited from listening to the music the way they want. The protection scheme has been cracked, and sites such as the Hymn Project[43] offer free software to convert protected AAC files to user friendly formats such as MP3 or Ogg Vorbis[63].

It is an option to legally bind the user to certain restrictions instead of technically restricting the rights to the material. Access to certain parts or services of the system can for instance require signing a legal contract. It will however be difficult to handle all possible variants of action, so this part of the system is problematic for a variety of reasons.

42 The file extension is m4a and m4p

43 The Hymn Project website: http://hymn-project.org/

9 Search Mechanisms

A archive or library of music and related information is of little use if the information and music cannot be found and retrieved. Searching and cataloging items may be based on descriptions and on the inherent properties of the objects. The descriptions of the data is called metadata. Some guidelines to metadata is outlined in 9.1. Search methods for music are described in 9.2.

9.1 Metadata

The metadata is the description of the data, or, in other words, data about data. For music recordings it might for instance include the title, the performer(s), the instruments, where it was recorded, who did the recording, what equipment was used, and so on. For digital objects IASA [3] points out the importance of having explicit, relevant and discrete records of all the technical details for the data creation and the record of changes. There are three methods for storing metadata in digital collections, and IASA suggests a combination of these [3]:

1. Inside the resource it describes.
2. Separate from the resource (external catalog).
3. Separate but linked to the resource.

[11] lists three common kinds of metadata:

- Descriptive metadata
- Administrative metadata
- Structural metadata

A thorough survey of what metadata should be included is beyond the scope of this thesis. However, standardizing the metadata may enable discovery of services in a distributed library system. The different library systems may for instance provide different search methods or access to different material. Standardizing discovery procedures means that all services may be accessed from one place without having to manually inquire each service. Protocols to support such discovery include

- Simple Object Access Protocol (SOAP) – a protocol used to exchange information.
- Web Services Description Language (WSDL) – describe the type of service, the location and the access protocol.
- Universal Description, Discovery, and Integration (UDDI) – Used to register services

The technology assumes that each library system is different and supports the interoperability of internally different systems. Users can find information on what, where and how to access information via Web Services. [64]

9.2 Music Search Methods

In addition to browsing an archive by different categories an archive can offer different searches. With powerful and efficient search tools, the value of the archive increases for the users as they can find what they are looking for faster and with less effort. Many different search methods exists. This chapter presents some possible methods.

9.2.1 Metadata Based Search

The first and most obvious way to search an database of music is to perform text searches for a certain song title or artist/composer maybe in combination with methods to narrow the search with genre, year etc. This type of searches typically uses metadata either from the file headers or metadata previously entered into a database for the music. The success of the search depends on how much metadata there is for each element, and that the user is able to remember the title, artist or some other relevant information.

9.2.1.1 Metadata Field Matching

The search can be done so that the user can specify which fields of metadata he wishes to search in, and specifies text to match the chosen fields. This can be very effective is the metadata is correct and the user knows what he is looking for.

9.2.1.2 Free Text Search

Free text search compares the search words with some or all of the data describing the item. A search for a particular word should then show hits regardless if it is found in the title, the lyrics or other places.

9.2.1.3 Visual Collaging of Music

People searching in a real library often uses some time looking and browsing through books or music manually. An approach to make this possible for a digital on-line music library has been made in a project described in [65]. The motivation for the project was that the observed behavior in a digital library was very different to that of a normal library. People using digital libraries used little time searching and looking through the material. Based on the assumption that the difference in behavior stems from that browsing a digital catalog is more cumbersome, the project developed a system that used visual collaging. It is based on “laid back” browsing which is based on recognition rather than the usual “sit forward” approach which is based on recall. Research shows that the human brain has a bigger capacity for recognizing things than to actively recall them. As an example, when learning a new language the active vocabulary is often much smaller than the vocabulary the person understands. The visual collaging was tested by displaying images of artists, CD-covers and other related pictures on the screen in a random order. After a while the image fades, and will eventually go away. New images are added at regular intervals. Figure 13 shows a screen shot of the prototype implementation (Image source: [65]).

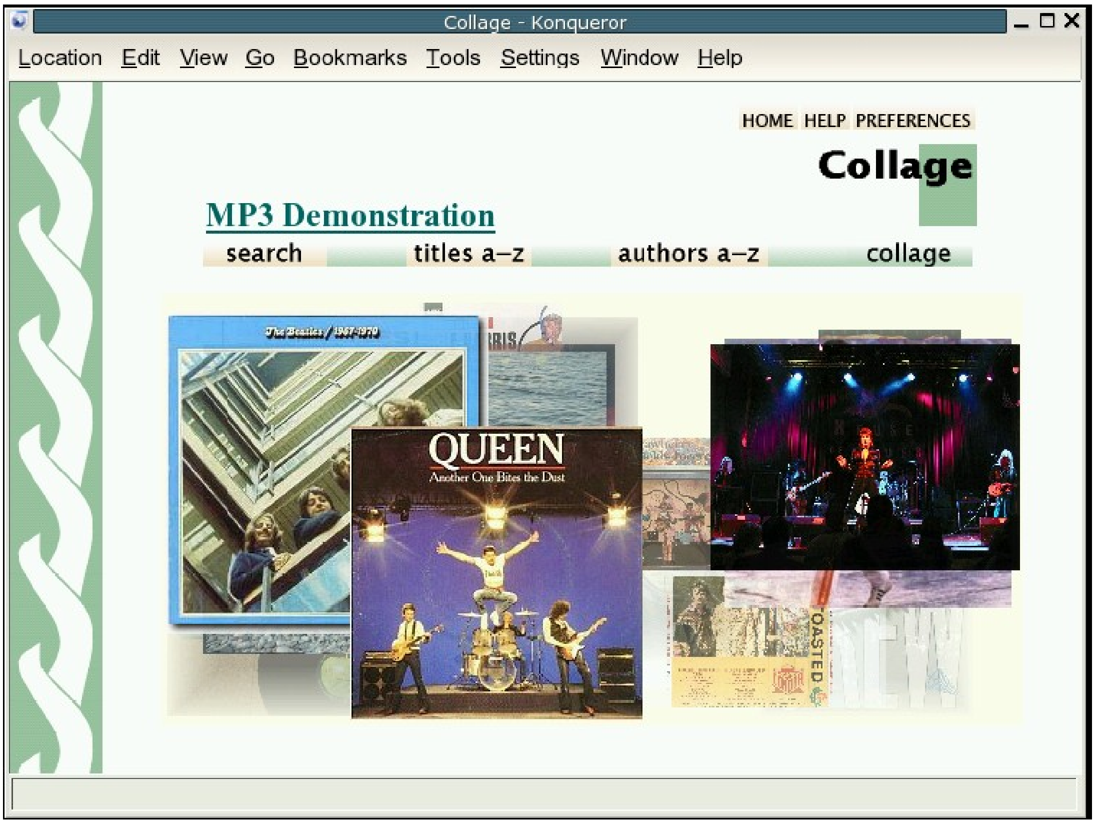

Figure 13: Visual Collaging of Music on a Web Page.

9.2.2 Content Based Query

Content based query are query methods that deals with melodic and rhythmic aspects of the music. Examples of queries might be:

- List all melodies that ends with a major C chord.
- Find a melody that is similar to this (the user gives the example in some way)
- How many songs have more than 70 percent eight notes?
- Query by humming – a person can hum or whistle a melody he remembers and the software finds similar songs for him. Can be useful for instance at a records store, as a computer system might be capable of "remembering" far more than a normal worker.

Being able to compare melodies can also be of scientific value. Musicologists can compare compositions of composers to see how much they have inspired each other, and how compositions are related. Using computers to do this instead of doing it by hand can be very time saving, and mean that such questions can be much easier resolved. Such query methods can also be used to resolve copyright issues. A composer can compare his newly finished work with a database of existing music to see if he can be accused of plagiarism before releasing it for sale.

These are aspects of the music normally not stored in the metadata of the music. Doing so would increase the workload of entering the metadata in manifolds. Instead, search methods have been developed where such queries can be performed directly on the stored data files. Search methods exists both for audio data and notated sheet music files.

In a survey about music information retrieval systems the authors lists the following (non comprehensive) music information retrieval tasks [66]:

- Copyright and royalties
- Detection of plagiarism
- Recommendation – find music of interest
- Sounds as – find music similar to a given example sung or uploaded by the user
- Mood – find music with a special mood
- Style – find music that belongs to a musical style
- Performer – find music performed by a particular artist
- Intertextuality – find works that are connected by using similar words or referring to each other
- Identification – find work by giving a example, for instance by humming
- Source – identify the artist or group of a recording when the metadata is missing

The survey is an attempt of mapping information retrieval systems to what tasks can be performed by it. The authors have found two dimensions that retrieval tasks can be mapped to. The first dimension is categories for the users of music information retrieval systems. Three groups are identified:

1. Industry – recording, broadcasting etc.
2. Consumers
3. Professionals – performers, teachers and musicologists

The other dimension is the level or detail of the retrieval:

- Work instance - the individual score or recording
- Work – objects that are basically the same, for instance two recordings of the same music with the same group
- Artist level – performer or creator
- Genre – Music that share a set of commonalities, for instance jazz, rock or baroque.

17 different systems have been mapped to these two dimensions, as Figure 14 shows. The study concludes that most music information retrieval systems operates at the work level. Tasks such as finding intertextuality in music, music recommendation and artist identification falls outside the operation of most of todays systems.

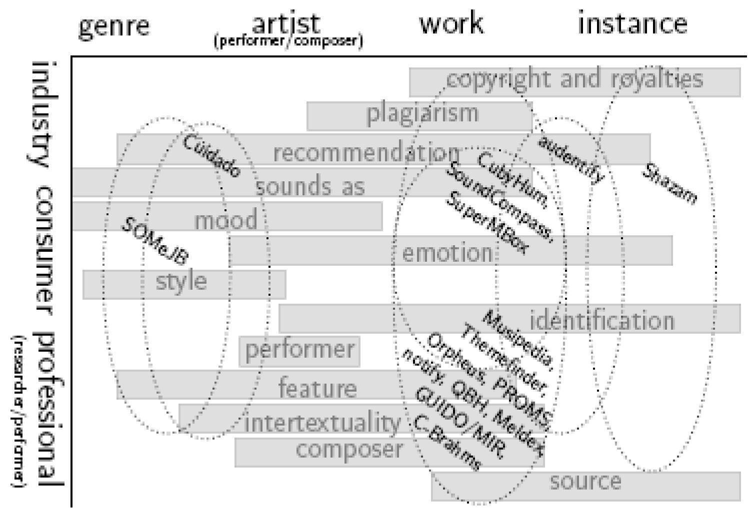

Figure 14: Mapping of music information retrieval systems. The vertical dimension is the main user groups, and the horizontal scale is the level at which the search systems work. The figure shows that most of the systems in the study operates at work level. (Source: [66])

9.2.2.1 Query by example

Most content based music information retrieval systems are variations of query by example. The main difference for the user is probably how the example he is searching for can be provided to the system. The systems are also using many different algorithms to perform the search, although this is normally hidden from the user.

Melodyhound (http://www.melodyhound.com/) is one web site offering content based search. It is the search engine used in Musipedia[44], and can search both sheet music from Musipedia and MIDI files on the web. All the search methods are based on the user giving an example of the melody he is looking for. The melodic or rhythmic example can be given in the following ways:

- Piano keyboard:
  The user can “play” the melody on a keyboard on the screen. As the user clicks on different notes, the user can see a picture showing the entered score and is also allowed to modify the score using the LilyPond language. (see Figure 10)
- Rhythm:
  The user can use the keyboard to tap the rhythm of a melody element ignoring tonal information.
- Melodic contour:
  The user describe the contour of the melody using Parson code. Parson code is a code describing the the direction of the pitch between to consecutive notes. The choices are up (U) , down (D) and the repeat (R), where the sign in the parentheses is the short

44 chapter 5.2.3

code. The coding ignores rhythm completely. A melody element can then be written as: UUDDRUDRUU. This is compared with a database of music where the Parsons code either have been extracted or manually entered. Although rhythm is ignored, and only basic melodic movement is considered, the method can separate between a number of tunes.

- Sing or whistle:
  The browser loads a Java program that can record using the PC microphone. When the user finish recording, the program tries to identify the pitch and duration of the notes. A picture showing the extracted notes is shown to the user, and he may add or remove notes as needed to improve the search.

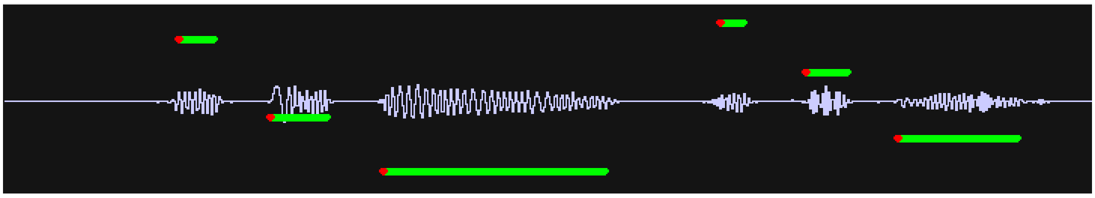

Figure 15: An example of a melody search with Melodyhound. After the user has sung or whistled a melody, the pitch is shown as red dots and the duration as green bars. The user may then add or remove notes and listen to the result to hear if it sound as intended.

- A related method of providing a melody is to upload or provide the location (URL) of an existing on-line MIDI file[45].

Themefinder (http://www.themefinder.org/) extends the list of searchable features further:

- Pitch – Letters are used to indicate the diatonic pitch name of the notes. The user can search for a particular sequence of pitches.
- Interval – A single or sequence of intervals can be searched for. The direction (up/down), quality (minor, major, augmented etc.) and size of the interval can be specified.
- Scale degree – Search based on the note's position on the scale. Similar to searching pitch, but wider searches can be performed as the search use the melodic key information to match identical melodies in different keys.
- Gross contour – Search using parsons code.
- Refined contour – A search method similar to Parsons code, but distinguishes between five instead of three movements. It separates between a small and a big leap up- or downwards between consecutive notes.
- The location of the melody element that is searched for can be specified to be in the beginning of the tune or anywhere within it.
- The key and mode can be used as search criteria.
- Meter can be specified. When specifying meter normal meters can be specified (2/4, 3/8, 4/4 etc.) in addition to keywords like: simple, compound, triple, mixed and so on.

45 An example can be seen at http://teuge.labs.cs.uu.nl/Rntt/mir/mir.cgi?pwd=none.

Part III The System

10 Summary of Needs and Possibilities

This chapter gives a summary of the needs of the digital archive system. Some possibilities are outlined, and the user requirements for the system are developed.

10.1 Terms

Document:	Audio, video, sheet music, text documents, metadata and other type of information stored in the archive. A document is not the same as a file as one document may be stored in several files. An example is a text document which constitutes one document, but may be saved as several files (pictures and text). Another example is a music book. The book is considered one document, but if there are many melodies all the individual melodies are also documents.

In this document the key words "**required**", "**recommended**", and "**optional**" in brackets or bold face are to be interpreted as described in RFC 2119.

10.2 Needs Revisited

Before specifying the concrete details of how the system shall be designed, it is in order to look back at the needs that the system shall cover. The functions the system shall perform was not entirely clear from the beginning of this project. The list of features and possibilities has grown as knowledge about the problem space has been built. The basic or fundamental features are those in the project statement. In addition to the stated problem, interviews with the initiators of the project and informal discussions with other persons interested has further increased this knowledge. Some current on-line systems have been evaluated in previous chapters to see if they are suitable as a solution. They have also served as examples and sources of ideas for this project.

10.2.1 Who is the System for?

The system can be of interest to the following groups of people:

- Researchers/musicologists
- Performers
- Composers/Editors of music
- Students of Music
- Teachers of Music

- General public

The system described here is mainly focused on the needs of researchers and students of music that belong to a university. The reason for this is that it will probably be the university that runs the system and have the economic responsibility. The advanced services of the system could therefore be provided mainly for university users.

Covering the needs of the general public to publish music and other material could soon prove to be a very comprehensive task, at least for audio and video. The base of users that use the system may grow very large, and the costs of operating it will also be much bigger. Persons wishing to publish their own compositions or garage band recordings can use existing services on the Internet. Several web sites selling sheet music can be used by regular persons to publish their music. The same goes for publishing recorded music. At web sites such as NRK Urørt (http://www11.nrk.no/urort/) Norwegian bands can promote themselves by uploading up to three songs that other users can listen to. They accept both video and audio files. Since there are already services on the Internet that is open for artists and groups to publish music, this system can be limited to offering this service to a smaller group of people.

On the other hand, parts of the system, for instance access to certain parts of the material and published information should be made available to the general public. A publicly accessible database of Norwegian compositions that otherwise is hard to find or even know about, could increase the interest in and use of such music. Work by composers that have been dead for a given amount of years is free of copyright issues, and access to such compositions can be free and unrestricted.

10.2.2 Project Statement Summary

From the project statement the main features of the system are extracted. The problem statement states the minimum of functions the system should perform.

> *The purpose of the system is to permit publishing of unknown, or unrecorded music in a way that makes it available to todays performers and researchers.*

This can be seen as two different functions, one to permit publishing, the other to permit access. Who is responsible for publishing is not specified, and this has been an area I have tried to look more into. Two different user groups that shall have access to the material is mentioned; performers and researchers. The unknown, unrecorded music is exemplified as sound, sheet music and possibly video.

From this the following can be stated:

- PS_1: The system shall permit the publishing of music in the form of sound and sheet music [Required]
- PS_2: The system may permit the publishing of music in the form of video. [optional]
- PS_3: The published material shall be available for todays performers and researchers [Required]

The next part of the problem states how the system shall be implemented:

> *The system shall have a central server with a access control system and user groups. It shall contain a user friendly library. The access control may also include payment for services.*

The first and last part can be stated as follows:

- PS_4: The system shall have central server. [Required]
- PS_5: The server shall have an access control system. [Required]
- PS_6: The system shall have different user groups. [Required]
  Remark: This is a direct consequence of the fact that the server shall have different user groups.
- PS_7: The central server shall maintain the user groups. [Required]
- PS_8: The access control system may include payment for services. [Optional]
  Remark: This means that access to certain services for certain user may be through payment. It does not imply that all users must pay for these services, so one possibility is that a document (as for instance a recording) is available to some users or user groups for free, while other users must pay to access them.

Next, the server shall contain a user friendly library. This will be split up into two requirements, since one is functional and the other is non-functional.

- PS_9: The server shall contain a library. [Required].
  Remark: This is a functional requirement, but it needs further clarification as it is not completely clear what is meant by the term library. It is interpreted here in the same way as it is in use in the rest of the thesis: the library is where the documents are stored, and access to the library means access to the documents in the library. Documents is the information stored in the system such as music (sound files, sheet music), video and other information (meta-information).
- PS_10: The library must be user friendly. [Required].
  Remark: This is a non-functional requirement which is highly subjective and not easy to measure in an objective way since it will vary from person to person what is perceived as user friendly. It can however imply that the library should be easy to use for people with little or non experience with such digital libraries.

The last part of the project statement says that it must be possible to include recordings and previously published compact discs in the library. This serves as an example of what the system should handle, but it is not a new requirement since it is it is already covered by PS_1.

10.2.3 Interviews

From the interviews in Section 2.3 the following system characteristics have been found:

- I_1: The system shall hold written information about the music (meta information). [Required]
- I_2: The system should function as a “record company” for universities and high schools. [Required]
- I_3: The system should function as a sound and sheet music archive.[Required]
- I_4: The system should be expandable to span several universities and countries.[Required]
- I_5: Material can be made available under restrictions such that only some users or user groups can access it.[Required]
- I_6: The system should support keeping track of the quality of published material.[Required]
- I_7: The system should support conference groups.[Required]
- I_8: Users can choose to stream music or download the music.[Required]
- I_9: The system should support generating audio previews of digitally notated sheet music.[Required]

Supporting I_9 does not necessarily mean that all notated music must have audio preview. If notated music is added to the archive as image files, making an audio preview may mean a lot of extra work. If notation files are added, the system should support generating MIDI files (or similar).

- I_10: Artists or groups can use the system to publish music.[Not recommended]

Although this was mentioned early in the process as a possible feature, we will not recommend it since this feature soon could use very much of the storage and processing capacity in the archive. There are other alternatives for artists and groups that wish to publish their music.

The next three requirements deals with user experience of the system. The system will function without them, but they add value to the system and can make it more exciting to use.

- I_11: The system should automatically generate lists of “best sellers”.[Recommended]
- I_12: The system should show the latest released material.[Recommended]
- I_13: Material listings can be sorted by category.[Required]

The next two requirements is ways to satisfy requirement I_6 since they are ways to keep track of the quality of the material. There may be other ways to achieve I_6, so we set these requirements to recommended and not required.

- I_14: Material can be commented by other users by logging in and comment or write an analysis of the material.[Recommended]
- I_15: Users can log in and grade material. The system should store the grades and generate lists of popular material.[Recommended]

The last requirement from the interviews is also set to recommended since the requirement is not necessary to make a archive system for audio and sheet music. Such information may be found elsewhere. It is probably a good idea to be able to link information about composers with their work, so we set this requirement to recommended.

I_16: The system should hold information about composers.[Recommended]

10.3 Other Issues

This chapter explains additional needs and features of the system that is needed. Some of the requirements are deduced from the requirements of the project statement and the interviews while other are ideas that I have developed while working with the thesis.

10.3.1 Quality of Service

To make sure that the system is not implemented in total regard to how the users experience it a demand for a certain quality of service is in order. The concrete details of the quality of service may be based on surveys of users.

- The system shall provide a specified quality of service [Required].

10.3.2 Search

To be an effective tool for spreading music among people it should be possible to search and browse through the data in effective ways. Both content based and metadata based queries should be supported. Section 9.2 sums up some of the possibilities of different search methods and it is clear that supporting queries of the type "find tunes similar to this" can be of use both for musicologists and other persons. Combining the metadata and content based queries should also be possible. The user can specify some information that must exist in the metadata such as a word in the title or an instrument name in combination with a melodic example. If the database of compositions is complete, it will then be possible to find out how many times Nordraak used a certain sequence of chords. To search in the contents of the music the music must be stored in a way that can be interpreted by computer algorithms. For audio the algorithms works directly on the audio data. Notated sheet music must be stored in a way that facilitates extraction of audio information.

This leads to the following requirement:

- The music (notated and audio) should be stored in such a way that content based search methods can be performed. [Recommended]
- It should be possible to make searches based on musical contents. [Nice to have] Examples: percentage of eight-notes, searches for similarities in melody or harmony etc.

In a full scale implementation of this system most search methods listed in Section 9.2 should be possible to implement. This gives the following recommendation, where the reader should refer to Section 9.2 for a explanation of the terms:

- The following search methods should be possible in a full implementation:
  - Metadata field matching [Required]
  - Free text search [Recommended]
  - Visual collaging [Nice to have]
  - Query by Example – On-screen Piano Keyboard [Nice to have]
  - Query by Example – Tap Rhythm [Nice to have]
  - Query by Example – Whistle or Sing a Tune in the Computer Microphone [Nice to have]
  - Query by Example – Melodic Contour Search – both using normal Parsons code and refined contour [Nice to have]
  - Query by Example – Pitch Sequence Search [Nice to have]
  - Query by Example – Scale Degree Search [Nice to have]
  - Query by Example – Interval Search [Nice to have]
  - Melodic Key and mode (minor, major, etc.) [Nice to have]
  - The meter should be searchable. [Nice to have]

10.3.3 Metadata

Metadata should comply with standards for digital libraries to ensure compatibility with similar archives so that it is possible to search through multiple archives at the same time. If the system is implemented at several different sites, a similar structure on the metadata makes it possible to interchange data between different systems. When a user wants to do a search he may then be offered to search all systems via the same interface instead of having to manually search each and every system. A web page at the Grieg Academy can then be used to search for items at for instance a university in Italy.

- Metadata should follow standards for digital libraries. [Recommended]

It should be possible to search for material based on terms like composer, style, price, name of the composition, artists, groups, instruments, year, etc. This means that the metadata should be able to hold information about all these items, and also that all are filled out when entering the metadata. However, there is a problem with having to fill in very extensive metadata information; it it labor intensive and must in most cases be done manually by humans. This must be weighted against the advantages of having extensive metadata when the archive is used.

It is harder to extract content information like key signature and time signature from audio files than from notated sheet music files. The sheet music may have tags specifying details about this, and this information can be extracted and used in searches. For audio files the key and time information can be included in the metadata description to make it possible to search in this information.

- Metadata for music documents should hold the key and time signature. [Recommended]

10.3.4 Contribution to the Sheet Music Archive

The labor involved in entering the sheet music can be spread across the user community by letting everybody contribute similar to Musipedia. There is of course a chance that material will be published that is not properly worked through or that the material has copyrights. To prevent this the administrators of the system should be able to remove material. Other users that discovers material which should not be published can then notify the administrators. The previously mentioned I_14 and I_15 also makes it possible to comment material, and material with very low quality can then be given low gradings.

This leads to the following additions to the specification:

- It should be open for anyone to contribute with notated sheet music and the accompanying metadata. [Required]
- Users with administrator privileges can remove documents from the archive. [Required]

10.3.4.1 Document Restoring

The system should also keep track of what documents each user has contributed with. It should keep a record of changes to each document with information about who changed it, and a possibility to restore the document to the previous version. This is a way to make sure that no single user can harm the system by destroying or changing documents permanently. If a user discovers that a document has changed and that something obviously is wrong, the document can easily be restored to its previous state. The question now is who should be responsible for restoring these documents. This is a question of how much trust can be given to general users and maybe this function should be reserved for administrators or a group of users with special privileges.

- The system must keep a change record of each document including enough information to restore the document to any previous state and information about which user performed each change. [Required]
- Users with administrator privileges can restore documents to any previous state. [Required]

10.3.4.2 Change Notification

To further improve the security of the system, it should be possible to let users subscribe to receive change information for selected documents. That way a user can be notified (for instance by e-mail) when someone has changed a certain document and the user can check himself to see if the change is an improvement or some sort of vandalism. A user would probably want to monitor documents he has contributed with himself. This can also be an yes/no-option when a new document is added to the archive. The notification should at least refer to the correct document, and may contain more information such as who did the change depending on the design of this function.

- A registered user shall be able to subscribe to be notified on e-mail for document changes to selected documents. [Required]
- When a new document is added, the user adding it may choose whether or not to receive change notifications. [Required]
- The notification should include enough information to identify the document in question. [Required]

10.3.5 User Database

To keep track of who has changed a document a user must register before changing documents. The system should store information about users so the same user can log in using the same username and password. E-mail address should also be provided, so that administrator can contact users. If a user forgets his log-in password, the e-mail address can also be used for a procedure to retrieve the password.

- Users must log in to edit, change or upload documents. [Required]
- The user database should at least hold a unique user name, password and the users contact information (e-mail address). [Required]
  Remark: The user name can be an arbitrary name chosen by the user. It does not need to be the users real name.

By combining information about the documents and the users it should then be possible for administrators to check out the history of users that have had reports of “bad behavior”.

The system should support some users with more privileges than other users. For instance some users may have rights to edit documents that other users only are allowed to view or must pay to download.

- The system should support user groups with different access rights to documents. [Required]

The access rights may for instance be just the ability to browse information, to download, to make new documents or to edit existing ones.

10.3.6 The Record Company Function

One of the features that is desired from the system is that it can function as an on-line record company for universities. This means that if a musicology research project includes making an audio production, this production could be offered for sale via the system. A possibility also exist to let different user groups have different access rights to this material. There are several options for how to charge for the material. Listed are four possibilities:

- Pay per track
- Pay per album
- Pay a time based subscription fee with free downloads
- Pay a time based subscription fee that gives a discount on album/track prices

This leads to the following requirement:

- One document can have different pricing alternatives for different users. [Required]

By charging for services it is necessary to make mechanisms that handles payment. This makes the system a bit more complicated, and to keep it as simple as possible (which is a good design rule), some assumptions are made about this system:

Access to sell material via the web-shop function should be limited. With the assumption that the amount of items for sale is very small compared to the overall number of documents, the control of this part can be centralized. Limiting what items can be sold means that items can be manually checked to see if they conform to a set of quality standards. This is a way to ensure that people are not fooled into buying low quality recordings by mistake. This again is in compliance with the demand that material offered on the network should be of a certain quality. The administration of normal user access is also simplified. When someone (like a professor) wants to sell some tracks he contacts the administrators which has privileges to set prices and advanced access rights for documents. An implementation of this can be done by include price/access information for each document in the database, but restrict access to alter price/access rights to administrators (or similar).

Restricting who can use the system to sell material also makes it easier to handle money transactions since the payments can be directed to only one or a limited number of accounts.

- There should be restrictions for which users may use the system to sell material. [Required]

Another issue concerning access to material is if some users shall have access to material from the web-shop without paying. A situation where this could be wanted is if the result of a research shall be free of use for the people involved in the project, for instance other people at the same department. This equals to set the price to zero for a group of users, and should therefore be possible after implementing the above requirements.

10.3.7 Archiving Function

The digital archive is supposed to be an archive for sound and sheet music material. This means that the system should store the material for an undefined (but long) time to make it available to future generations. IASA [4] notes that no digital preservation system in itself is a permanent storage solution. Rather it should support change of technology so that transfer from one storage system to a new and better one is possible. If it is not possible to upgrade or replace the system the whole collection can be at risk.

- The storage system should store the material securely for a long time frame [Required]
- The storage system should support transfer of the material to another storage system [Required]
  Remark: This must be done before the old storage system stops functioning or loose its technological support.

Guidelines for managing long-term storage of audio material can be found in the IASA "Guidelines on the Production and Preservation of Digital Audio Objects" [4] [46].

10.3.8 File Constraints

As section 3.1.2 mentions, the audio should be stored with high quality so that the possible use scenarios of the audio in the future is not restricted. This leads to the following demands:

- Audio should be stored in a format that keeps all audio information.[Required]

When audio is downloaded/streamed the audio data can be sent in a compressed form depending on what quality the user needs. The user can for instance be given choices between high and low compression and raw audio data.

- The archive system should store a master copy of each document and offer copies to users at the same or lower quality. [Required]

One design choice here is whether the lower quality files should be generated once and stored on the server or be generated on the fly when a user request it. Both choices user server resources, the first will consume more disc space while the other use more processing power when the requests are processed.

The same goes for video, where the quality of the recorded video should not be reduced before it is stored in the archive. It is possible to make low-resolution copies using conversion algorithms if the videos are to be offered to users with low bandwidth or devices with small screens. This can be done while the original file still lies intact in the archive.

- Video should be stored in a format that keeps all video and audio information. [Required]

Next; material that is born digital should be stored in the original resolution. This follows from wanting to preserve all information. Since digital material can be copied exactly bit by bit, this is not to complicated technically. However, it means that lossy compression should not be used on the original videos.

- Material that is born digital should not be bit-reduced. [Required]

46 pp 50-51.

Sheet music file types has already been mentioned earlier and is not repeated here.

The system should also handle other file types than just the specified audio, video and notation formats. It should be possible to add new file types after the system has been put into use.

- The system should support adding other files than audio, video and sheet music files. [Required]

10.3.9 Needs Pertaining to User Experience

In addition to searching the archive it should be possible to browse the material by composer, title, genre and possibly by which user published the material.

- It should be possible to browse the material by composer, title, genre and possibly by which user published the material [Required]

When sheet music is used in reports to underline a musical concept or when just browsing scores to decide whether or not to use the music, it will be an advantage to be able to listen to the score directly.

- There should be a possibility to listen to sheet music directly from the user interface. [Recommended]

Changing the pitch and speed of audio music is possible with many sound processing software systems. Programs such as Amazing Slow Downer[47] allow realtime change of pitch without changing the speed of the playback and vice versa for audio files and CDs. Figure 16 shows the program with controls for speed, pitch, volume, start point, end point and an equalizer function. Embedding such functions into the user interface of the archive system would be a nice feature.

Figure 16: Amazing Slow Downer is an example of software that lets the user change playback pitch and speed independent of each other.

- There should be support for changing pitch and speed of the audio playback [Nice to have]
- Sheet music containing multiple voices should support regulating the playback volume of the different voices individually [Nice to have]

This can for instance be used for just hearing one of the voices when rehearsing.

- When notated sheet music contains multiple voices the system should support printing a selection of the voices individually. [Nice to have]

For the on-line store, there should be support for previews before you buy. For sheet music, the preview could be the first page of the music possibly combined with a way to listen to the composition. For audio recording and video, the preview could be the first minute or so of the recording in low resolution, for instance compressed in MP3 at 64 kbit. This way it is less probable that the user will be satisfied with the preview alone. It will rather function as a sample for the real product and can therefore contribute to a higher sale.

- The on-line store should offer a preview of it's products. [Recommended]

10.3.10 Language and Interoperability

- The system should be made in such a way that it can interoperate with similar systems at other universities. [Recommended]

This means that the system should follow standards which enable interoperation. This includes standards for document storage and metadata so that metadata information can be searched across different systems at the same time. A well designed interface for searching the system can facilitate searching more sites at one time.

One potential problem with inter-operation across different countries is language. Although the system presents information to the user in his native language it should be possible to search in catalogs from universities that use other languages. For instance a French user will want a French user interface. Searching for artist names, titles and the search by singing (etc.) should be possible regardless of the language since for example the artists name will still be the same.

- Language: it shall be possible to use the system and perform searches etc. in different languages. It should be possible to add other languages after the system has been put into use. [Recommended]

10.3.11 Provide Information

There are many possibilities for generating information about the contents of the digital archive to the users. Some possibilities are:

- The system should provide a list of best-sellers or the most popular downloads [Nice to have]

Users can also be given the opportunity to give reviews of a composition or other material. This can be in the form of an extensive written review, a voting or grading mechanism or just a short comment.

- Users may log in and give comment, review or vote to material [Nice to have]

If this is implemented it lead to:

- The system shall show comments, reviews and votes for the selected document [Nice to have]

10.3.12 Copyright Issues

- Material with specific copyrights should only be available to the intended users. [Required]

10.3.13 Quality of the Material

If the archive system is supposed to be a place where research work can be published the users must be convinced that the quality of the material is the best. Perhaps there should be a clear separation of what is research publications and the other material. For material which users must pay to download/view there should also be some quality check so that people are not fooled into buying low quality material unknowingly. This can be the technical quality of the recording (amateur, home studio, professional studio) and some kind of measure of artistic quality.

This means that some authority can be responsible for the quality of published research work and CD-recordings. Other material can be based on user feedback schemes mentioned above in 10.3.11 which is one way to give a quality measure of the material. This can for instance be used on the material which is not of research standard.

- There should be ways to ensure that the material published as research has a certain quality [Recommended]

10.3.14 Video

The ability to handle video as a part of the system has been suggested in the interviews. Examples where this is useful is for instance research on how to play on an old instrument. It is possible to spread information about techniques by recording it on video. Public concerts held at universities, for instance exam concerts, may also be something a university will want to publish.

- The system should be able to store and present videos. [Required]
- The system should support streaming videos in compressed resolution. [Required]

What constitutes an adequate resolution for streaming depends on what the video shall be used for. In many circumstances a compressed video can be adequate for the user. It saves bandwidth on the network and it takes less storage space if the user downloads the file. All users may however not want compressed video. Compressed video may hide the necessary detail to make a proper judgment in for example a photo-finish to decide the winner in a dogs race.

10.4 Security Policy

The digital music library should have a security policy. Conallen [67] suggests at least to have a policy for the following security issues:

- ✗ Who will have access to the system?
- ✗ What time can they access the system?
- ✗ What are they allowed to do with the system?
- ✗ How are users added to the system and how are they removed?
- ✗ Surveillance of the system: What is logged and how often are the logs checked?

10.4.1 Secure Transfer of Data

To secure the transfer of data between the server and the client against tapping, the data can be sent using encryption. One method to achieve this is to use Secure Sockets Layer (SSL). SSL is an encryption method introduced by Netscape. It encrypts the higher level protocols such as HTTP and FTP. It relies on the server showing a certificate from a Certificate Authority (CF) to identify itself for the user. SSL is widely used on the Internet to support secure e-commerce[67].

11 Specification

This chapter outlines the specification of the system. It is based on the needs and discussions in previous chapter, especially Chapter 10. Explanation of the background for the requirements can also be found in Chapter 10. The requirements are mostly listed in the same order as found in Chapter 10 except that they are sorted into functional and nonfunctional requirements.

11.1 Functional Requirements

This section lists the functional requirements found for the system. Functional requirements are statements that say what services the system shall provide, how the system should react to input and statements concerning the behavior of the system in various situations.[68]

11.1.1 Functional Requirements From the Project Statement

PS_1: The system shall permit the publishing of music in the form of sound and sheet music [Required]
PS_2: The system may permit the publishing of music in the form of video. [optional]
PS_3: The published material shall be available for todays performers and researchers [Required]

The following four requirements are both functional and nonfunctional. They are therefore listed also under nonfunctional requirements.

PS_6: The system shall have different user groups. [Required]
PS_8: The access control system may include payment for services. [Optional]

11.1.2 Functional Requirements From the Interview

- I_1: The system shall hold written information about the music (meta information). [Required]
- I_2: The system should function as a “record company” for universities and high schools. [Required]
- I_3: The system should function as a sound and sheet music archive.[Required]
- I_4: The system should be expandable to span several universities and countries.[Required]
- I_5: Material can be made available under restrictions such that only some users or user groups can access it.[Required]
- I_6: The system should support keeping track of the quality of published material.[Required]
- I_7: The system should support conference groups.[Required]

I_8: Users can choose to stream music or download the music.[Required]
I_9: The system should support generating audio previews of digitally notated sheet music.[Required]
I_10: Artists or groups can use the system to publish music.[Not recommended]
I_11: The system should automatically generate lists of "best sellers".[Recommended]
I_12: The system should show the latest released material.[Recommended]
I_13: Material listings can be sorted by category.[Required]
I_14: Material can be commented by other users by logging in and comment or write an analysis of the material.[Recommended]
I_15: Users can log in and grade material. The system should store the grades and generate lists of popular material.[Recommended]
I_16: The system should hold information about composers.[Recommended]

11.1.3 Functional Requirements from Chapter 10.3 "Other Issues"

- The music (notated and audio) should be stored in such a way that content based search methods can be performed. [Recommended]
- It should be possible to make searches based on musical contents. [Nice to have]
- The following search methods should be possible in a full implementation:
- Metadata field matching [Required]
- Free text search [Recommended]
- Visual collaging [Nice to have]
- Query by Example – On-screen Piano Keyboard [Nice to have]
- Query by Example – Tap Rhythm [Nice to have]
- Query by Example – Whistle or Sing a Tune in the Computer Microphone [Nice to have]
- Query by Example – Melodic Contour Search – both using normal Parsons code and refined contour [Nice to have]
- Query by Example – Pitch Sequence Search [Nice to have]
- Query by Example – Scale Degree Search [Nice to have]
- Query by Example – Interval Search [Nice to have]
- Melodic Key and mode (minor, major, etc.) [Nice to have]
- The meter should be searchable. [Nice to have]
- Metadata for music documents should hold the key and time signature. [Recommended]
- It should be open for anyone to contribute with notated sheet music and the accompanying metadata. [Required]
- Users with administrator privileges can remove documents from the archive. [Required]
- The system must keep a change record of each document including enough information to restore the document to any previous state and information about which user performed each change. [Required]
- Users with administrator privileges can restore documents to any previous state. [Required]

- A registered user shall be able to subscribe to be notified on e-mail for document changes to selected documents. [Required]
- When a new document is added, the user adding it may choose whether or not to receive change notifications. [Required]
- The notification should include enough information to identify the document in question. [Required]
- Users must log in to edit, change or upload documents. [Required]
- The user database should at least hold a unique user name, password and the users contact information (e-mail address). [Required]
- The system should support user groups with different access rights to documents. [Required]
- One document can have different pricing alternatives for different users. [Required]
- There should be restrictions for which users may use the system to sell material. [Required]
- The storage system should store the material securely for a long time frame [Required]
- The storage system should support transfer of the material to another storage system [Required]
- Audio should be stored in a format that keeps all audio information.[Required]
- The archive system should store a master copy of each document and offer copies to users at the same or lower quality. [Required]
- Video should be stored in a format that keeps all video and audio information. [Required]
- Material that is born digital should not be bit-reduced. [Required]
- The system should support adding other files than audio, video and sheet music files. [Required]
- It should be possible to browse the material by composer, title, genre and possibly by which user published the material [Required]
- There should be a possibility to listen to sheet music directly from the user interface. [Recommended]
- There should be support for changing pitch and speed of the audio playback [Nice to have]
- Sheet music containing multiple voices should support regulating the playback volume of the different voices individually [Nice to have]
- When notated sheet music contains multiple voices the system should support printing a selection of the voices individually. [Nice to have]
- The on-line store should offer a preview of it's products. [Recommended]
- The system should be made in such a way that it can interoperate with similar systems at other universities. [Recommended]
- Language: it shall be possible to use the system and perform searches etc. in different languages. It should be possible to add other languages after the system has been put into use. [Recommended]
- The system should provide a list of best-sellers or the most popular downloads [Nice to have]
- Users may log in and give comment, review or vote to material [Nice to have]
  If this is implemented it lead to: The system shall show comments, reviews and votes for the selected document [Nice to have]

- Material with specific copyrights should only be available to the intended users. [Required]
- There should be ways to ensure that the material published as research has a certain quality [Recommended]
- The system should be able to store and present videos. [Required]
- The system should support streaming videos in compressed resolution. [Required]

11.2 Nonfunctional Requirements

Nonfunctional requirements are constraints to the system or its services.[68]

11.2.1 Nonfunctional Requirements From the Project Statement

PS_4: The system shall have central server. [Required]
PS_5: The server shall have an access control system. [Required]
PS_6: The system shall have different user groups. [Required]
PS_8: The access control system may include payment for services. [Optional]
PS_7: The central server shall maintain the user groups. [Required]
PS_9: The server shall contain a library. [Required].
PS_10: The library must be user friendly. [Required].

11.2.2 Nonfunctional Requirements From the Interviews

- The system shall provide a specified quality of service [Required].
- Metadata should follow standards for digital libraries. [Recommended]

11.2.3 Other Nonfunctional Requirements

Other nonfunctional requirements are for instance timing constraints, reliability properties and response time. These constraints are not described in this document. They should be found before the system is implemented. The following items should at least be taken into consideration:

11.2.3.1 Usability

It is important that the system is usable for the users. It is not straightforward to define what is a usable system. This archive system will have many different types of users, and the proficiency of the users in handling computers and software may vary to a great extent.

11.2.3.2 Performance and Robustness/reliability

The performance properties depends on how many users shall be served and many other properties. The exact performance and reliability properties remains a subject for further studies.

11.2.3.3 Security

- The system should provide a secure storage of cultural heritage material. [Required]
- The system should store user data safely.[Required]

Other security requirements will probably be found after further studies as suggested above.

12 Design

This chapter outlines the design of the system. One of the main motives for the design is that that the specification should be a useful digital library system for Universities that teach and do research in music.

First a distributed architecture is described. Then the access control system for the system is discussed and a user access model is developed. The next sub-chapter deals with how existing software and file formats can be used in the system. The last part explains the development of the network of digital music archives.

12.1 Distributed Architecture

Although the project statement says that the system shall have a central server system it is also seen that the system should support being extended over several universities and countries. This may be done by having one server handling the user requests. The server then use remote resources via protocols that support a distributed architecture as shown in Figure 17.

In a distributed client-server environment the users web client can generate a query. The query is sent to an agent (a machine) that handles the requests and knows what other machines can be contacted to perform the search. The agent contacts other agents and puts together the full report which is then sent to the user. By standardizing all communication interfaces different systems will be able to talk with each other. More elaborate examples can be found in the Greenstone papers [51] & [53].

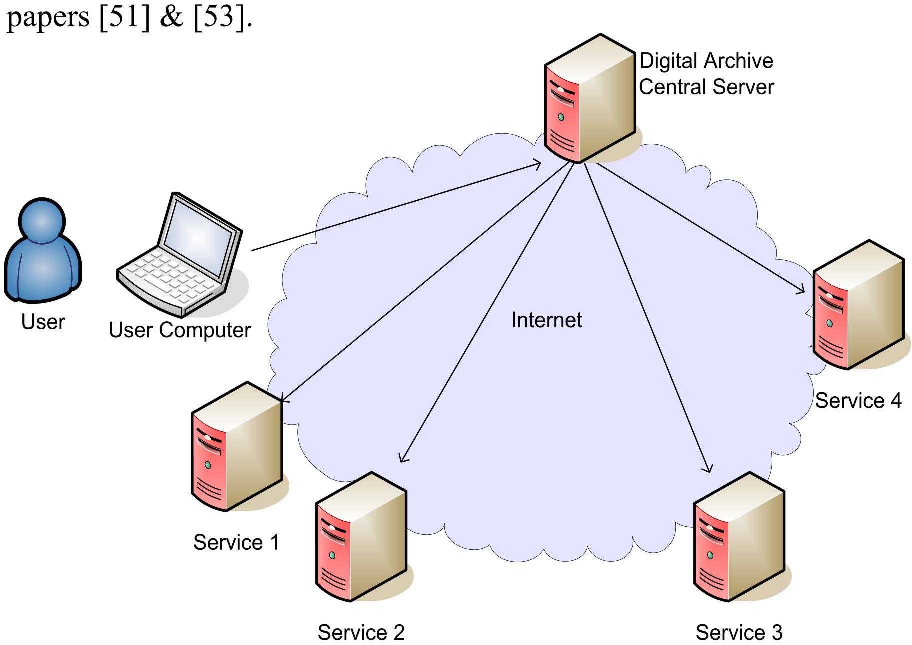

Figure 17: A distributed architecture is shown where the Digital Archive Central Server handles requests from the users. The Central Server may in turn use other services placed elsewhere in the network to fulfill the request.

The user interface can be a normal web page on a web-server. File download should be based on TCP/IP which gives correct delivery of data from the server to the client. Streaming can be based on the RTP -protocol (Real-time Transport Protocol). RTP splits the media stream in RTP-packets that are sent over other protocols such as UDP/IP [69]. The reasons for not using TCP for streaming are:

- TCP retransmission: When TCP detects packet loss it will retransmit until the packets are successfully received. This introduces a time delay. When viewing a video or audio stream the timing is normally more important than the effect of loosing one frame[70].
- Window size: When TCP detects packet loss the window size is reduced. TCP then uses slow-start mechanism to regulate the windows size. This means that less video can be transferred per second after a packet loss until the windows size is large enough to handle the stream[70].

UDP is a stateless protocol that offers no guaranties regarding packet loss and the order in which they arrive. UDP also has less overhead than TCP. This makes UDP better suited for real-time applications where packet loss can be tolerated.

RTP offers no Quality of Service. The application which use RTP can still check for packet loss because RTP offers a numbering of the individual stream packets. It is then up to the application to decide what to do when is experiences packet loss[69]. RTP over UDP can therefore be used for the streaming part of the digital library system.

A protocol called RTSP can be used to control the media stream. It functions as a remote control for the media stream and defines commands such as play, stop, pause, rewind and fast forward from the client to the media stream server. RTSP is not a media stream itself, instead it controls the correct timing of media streams from the media server to the client[71].

When a user wants to view a video from the archive the web browser can load a Media Player. The Media Player is normally a plug-in to the web browser such as QuickTime. Then a HTTP request is sent to the Digital Archive Web Server and the server response is a description of the media file. The media file description is sent to the Media Player which sets up the media stream with the Media Streaming Server using RTSP. Figure 18 shows an example streaming session which uses RTSP to control the media stream. The example is from [72].

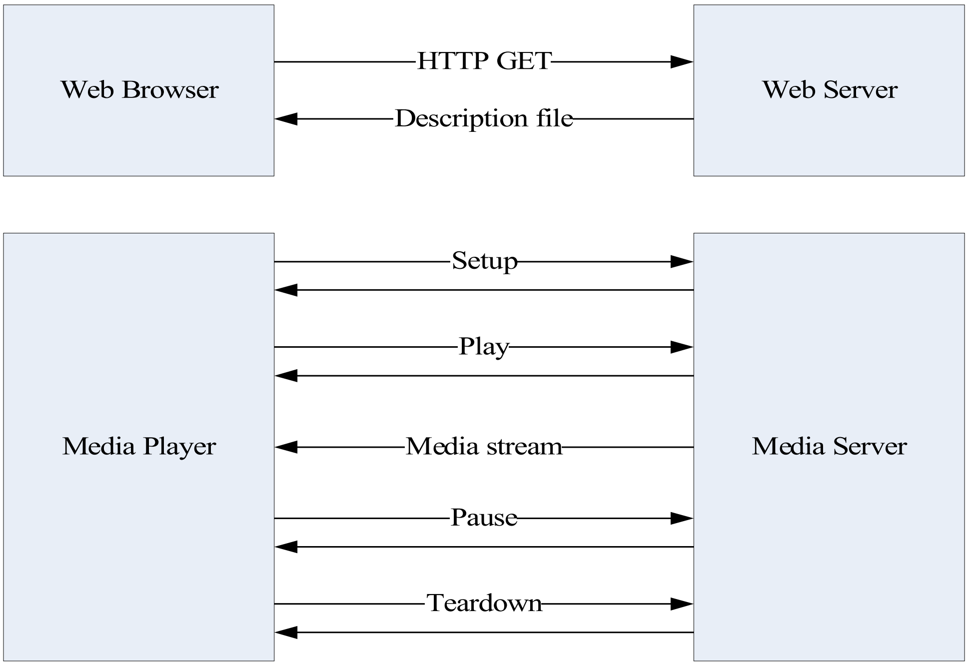

Figure 18: Example streaming session using RTSP.

12.2 Access Control and User Groups

In this section a database model of the access control system is developed. First the users of the system are explained. Then the different aspects of an open contra a closed system are presented. The access model that then is introduced is a combination of a closed and an open system. Several scenarios for a digital library system are modeled using the access control database model to show that the model can support many different areas of use.

12.2.1 Primary Actors

The primary actors of a system are the persons that have user goals fulfilled by using the services of the system. Here, the primary actors are called users, the operators and the system providers.

- The users of the system are persons that use the system for publishing and those that access the published material. There may be different access rights for different users of the system.
- The operators are responsible for operating the system at a certain university, for instance at the Grieg Academy in Bergen. The operator is both responsible for the artistic quality of the contents and for the technical equipment the system runs on.
- The system providers are responsible for connecting new universities (operators) to the network, and are responsible for the interoperation of the system. The system

provider may also be an operator, for example if the system is only realized at one location.

12.2.2 User Access Models

One of the main choices in the design and specification of the system is access. The choices range from a totally open collaboration project with no login and user groups to a closed system with rigorous access control systems and advanced authentication and authorization schemes.

12.2.2.1 Open Model

A open model means that all users are free to view, download and contribute to the collection. This means that control of the system is highly deregulated and spread. There is no central controlling authority.

Advantages of such a model is that the collection of available sheet music that in reality have been unavailable to most people may grow large faster than when only a few people contributes. Making the contents available with few or no restrictions may also encourage people to use old compositions. It is probably not unrealistic to imagine that more people will use a free product than one that cost money or the inconvenience of registering. However, the downside of a totally open solution may be hazardous: - A unfriendly user can destroy the contents of the archive, or (perhaps worse), alter the contents so that users are unaware that they have downloaded a product with errors. This should be avoided. A totally open model is perhaps not a good model for safe preservation of cultural heritage, as long as the safety is dependent on the goodwill of the users.

The Open Model has a self regulating control mechanism, as the users themselves can control each other. If something is deleted, another user can reverse the action (if the system allows actions to be reversed). Similar, if a user discovers an error in an transcription, he can edit it and let the new edition be available.

12.2.2.2 Closed Model

In the Closed Model access to all parts of the system is restricted. It represents the opposite of the Open Model. Users must register with some authority before they are granted access to the system. The registration can include signing a contract with terms of use. In the Closed Model the control is centralized.

Tight control from some central authority and strong regulation of what the users may do to the system means that it is easier to regulate the security. Only the central authority may submit, alter or delete the contents.

The system is not open for access without registration. It is also possible to only give access to certain parts of the contents for different users, perhaps by grouping users into user groups with defined access rights.

Disadvantages of the Closed Model is that the contents is only available to users that register. If there are rules for who are allowed to register like that a person must be associated with a university or music institution, a number of persons will be prevented from using the material. This also means that fewer people are allowed to contribute to the archive, and the growth may be slower than in the Open Model.

12.2.2.3 Combination Model

As mentioned, the Open Model and the Closed Model represent two very different approaches to modeling access. The main differences of the two models is shown below in Figure 19.

CLOSED MODEL	***versus***	***OPEN MODEL***
Security		Freedom
Centralized control		Decentralized control, self regulating
Controlled submission		Free collaboration and submissions
Rights management Access costs Membership		Free access Free use
Closed user community		Unrestricted user base

Figure 19: The Closed versus the Open Model for user access

Both choices have their advantages and disadvantages. One way to design the system is to try to combine the best possibilities of the Open and the Closed Model. We call this the Combination Model:

CLOSED MODEL	***versus***	***OPEN MODEL***

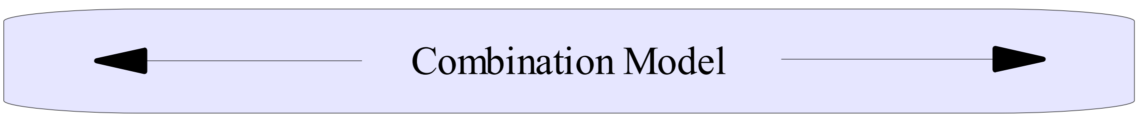

Figure 20: The Combination Model combines the features of an open and a closed model.

In the Combination Model, some services and actions can be restricted to special users and user groups while some are free and unregulated. It tries to take advantage of the best sides of both the closed and the open model. Examples of this follows in the next sections.

12.2.3 Collections

Before continuing the specifications of user access we shall look at the concept of a collection.

When this archive system is in use one can imagine that at any time there are some documents that are finished and should be searchable. Other documents are unfinished and people are collaborating to finish them, some documents have a price for one group of users but are free for other and so on. A document can be said to bee in a state that depends on which user tries to access it.

One way to handle this is introduce the concept of a collection. A collection is one or more documents that share a set of properties. One document may belong to several collections, and one collection may be a subset of another collection. The collection keeps pointers to all document that belong to the collection so that a document belonging to several collections need only be stored at one place. This prevents different versions of the same document from appearing in the archive and saves storage space. The collections can be public or private or shared between a limited set of users. Private collections are just visible for the creator of the documents and may be unfinished documents et cetera. A collection can also be set to be public readable, but not not editable by others than the owner.

A user may have access to one or more collections. For instance, all users should have access to the public collection for browsing without logging in. Submissions to the public collection could be made after logging in.

Following is an example of how the collection system works. At first Herbert, a regular user, is working on a document that belongs to his private collection. In the figure below this document is called “5”. When he wants to share it, he can put it in the public collection, after which it no longer belongs to his private collection. It is then shared with the whole world. He could also choose to only share it with his old friend Peter, which also is a user in the system. A new collection is then made called Shared_H_P (other names could be used). The Shared_H_P collection then points to document “5” and both Peter and Herbert has access rights to the collection.

The next figure shows a small example of the collection concept:

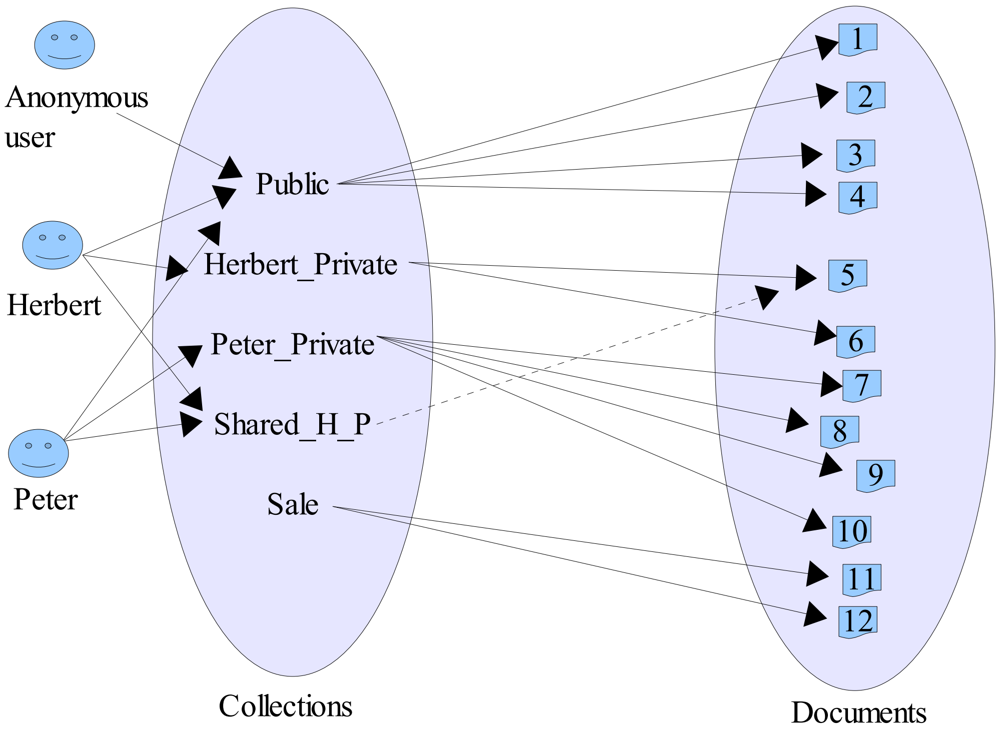

Figure 21: A example of the relationships between collections and documents. The arrows indicate access rights. Some documents are public, some belong to Herbert's and Peter's private collections and some are for sale. Document 5 is created by Herbert, but is later shared between Herbert and Peter, accessible through the collection called Shared_H_P.

12.2.4 Access Management Model

The following specification is one possible solution to the access management for a digital archive and library system. It is based on the requirements in Chapter 10 and is a recommendation for how the system can be designed.

Users need not to log in to browse and download public material. Access to other parts of the system than downloading and browsing the sheet music archive may be regulated by registering users.

After registration the user can be a member of one or several groups. Each group have defined access rights in the system. The system should support that the rights of groups and users may change, and should support changing the rights while the system is running.

Figure 22 below shows one way to map documents and users to each other. It is a simplified model, and does for instance take not into account details about metadata. In the model, a User is a person that somehow is registered in the system. If a user want to use the system without registering, he will automatically be registered as “anonymous”. “anonymous” is then one instantiation of the entity class User. When users of the system register, they will be other instantiations of the class User.

A User may be a owner of a Document via the relation “Owns_document”. A Document only has one owner, but one User may own many Documents.

Documents may share a number of properties concerning who has access to them, and documents can therefore be grouped into Collections. One Document may belong to several Collections, and one Collection may have several Documents. The relation between them is called “Belongs_to_Collection”.

Several Users may share the same properties and access rights to documents. Instead of having to set these properties for each user, we add the class Group. A Group is a group of users which share a set of properties. A User may be a member of many Groups, and a Group may have many Users. The relation between them is called “Member_of”. The Group was not shown in the example in Figure 21 above.

To let user groups have access to collections, a relation between Group and Collection called “Has_Access_To” is added.

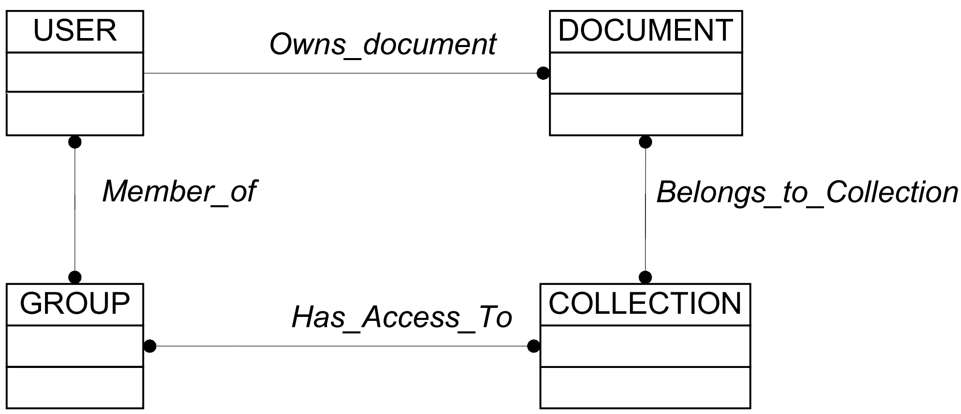

Figure 22: UML class diagram showing the relation between User, Group, Document and Collections.

This design should be able to support many different access schemes as is shown in the following examples.

12.2.4.1 Example 1: Free Collection

Imagine that some old sheet music, a music book with 20 waltzes that were collected 150 years ago, is digitized and supposed to be made publicly available. First, it is a question whether each single piece of music shall be one document or if the whole music book shall be one document. Both ways might work, but for this case we let the note book be a Document called “1856_music_book”. The scores and other information is mapped from the “1856_music_book” -document to the individual documents with the individual waltzes.

After the document is registered, a relation to an existing Collection called “Public_sheet_music” is made. The “Public_sheet_music”-collection has a relation to an existing Group called “Default_group”. The “Default_group” is a group that all users automatically is registered with. All collections that is intended to be freely available for search, download and use can be made available for this group. What kind of access the group is allowed to do is a attribute of the “Has_access_To”-relation. Figure 23 shows how the new music book relates to the default group.

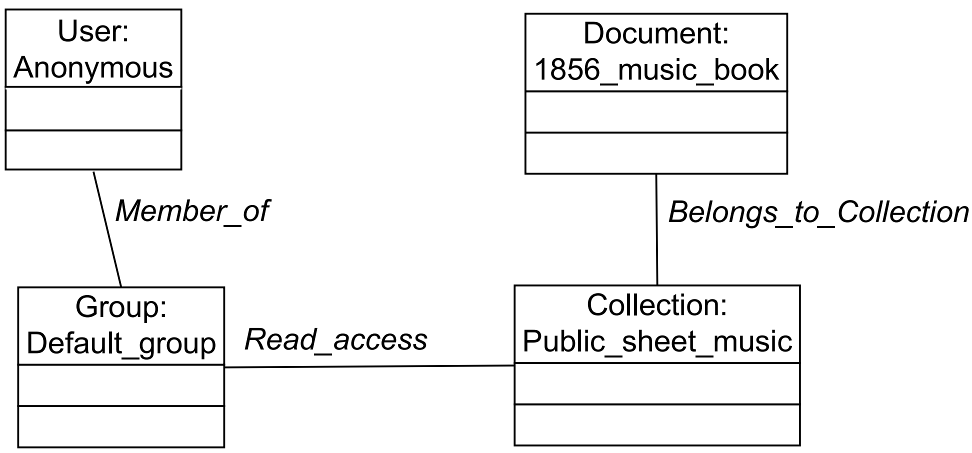

Figure 23: Diagram of a music book that shall be made publicly available by relating it to the “Default_group”. All users are members of the Default_group.

Now all users have access to the music book. As mentioned it must also be specified what it means to have access. The relation “Has_Access_To” should have an attribute that describes the type of access. The access choices can be:

- Search – The collection is searchable and browseable for the user. This means that the user group have access to information about the documents in the collection. It does not mean access to the documents themselves. Where to draw the line between information relevant for search and browsing and what information constitutes the document itself should be further specified.
- Read – The user group is allowed to read the documents in the collection. This means access to download or stream the document.

- Write – The user group is allowed to edit documents in the collection. This include making new documents, deleting old ones and making changes to existing ones.

12.2.4.2 Example 2: Record Store

The next example shows how the “record store” may work within the model. A record store should let anonymous users browse and search through the collection, otherwise it is impossible to know what the store offers. Since one document may belong to several collections at the same time it is possible to let one user group have full access (read/write) to the documents while other groups may only search them.

One document can have different prices for different user groups, which means that the price information should not be a static value for the Document object. It should rather be an attribute of the relationship between a Document and a Collection. We therefore introduce the attribute “price” to the relation “Belongs_to_Collection”. For the sheet music collection in Example 1 the price is simply set to 0, and no payment procedures are taken.

Imagine the following situation: A research report (document) is supposed to be publicly available for a price of NOK 200,-. The report shall be freely available to employees at NTNU. No one else than the owner shall be allowed to edit the document.

A collection used for public sales called “Public_sale” already exists in the system. This collection is browseable for all users (the “Default_group). The research document is added to the system and given a identifier, for instance “Research_report_200”. A relation to the “Public_sale” Collection is made, at the price attribute is set to “200 NOK”. Luckily an additional group also exists in the system which is the group consisting of the employees at NTNU. The group is called “Employee_NTNU”. To make the “Research_report_200” available to “Employee_NTNU” a new Collection is added to the system (if it does not exist) which is called “NTNU_free_download_for_employees”. A relation from the “Research_report_200”-document to the “NTNU_free_download_for_employees”-collection is added with the price attribute set to zero.

Figure 24 illustrates the situation. Two users, Peter and Herbert, wants to access the same report. Herbert, who works at NTNU, is registered both to the “Default_group” and the “Employee_NTNU” groups. He has access to the report through both memberships, and the system automatically uses the membership where the report is cheapest. He can then print and read the report knowing that he did not spend a dime on it. Peter on the other hand, is not an employee at NTNU. The only path from Peter to the report is via the “Default_group”. He will then have to pay 200 NOK to download the report.

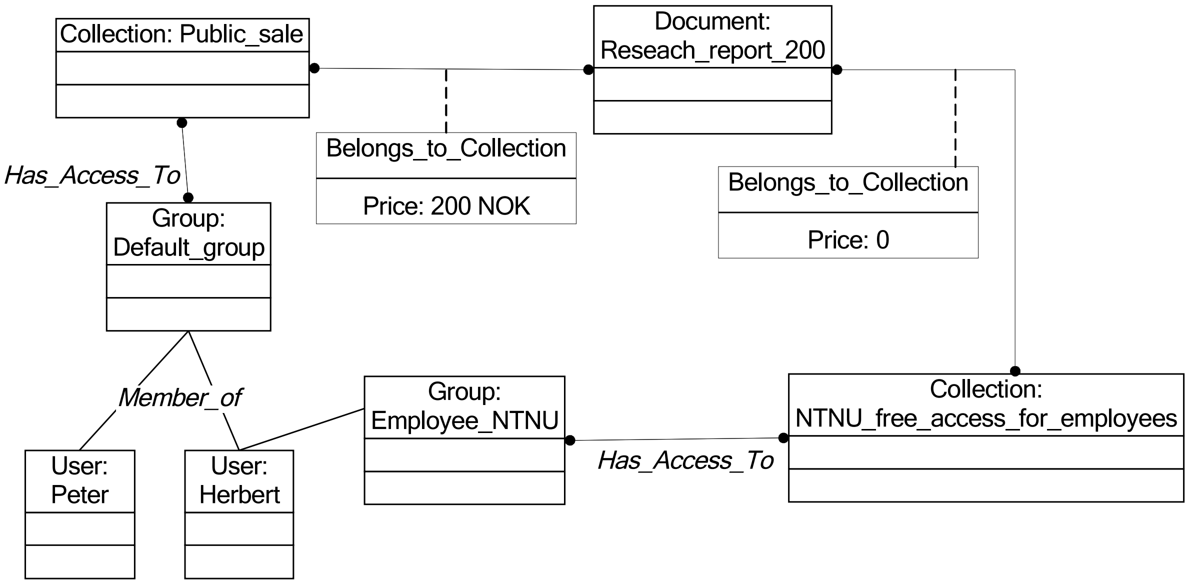

Figure 24: The same document may have one cost for one user group and another price for another user group by assigning the document to more than one collection. Here, Peter must pay 200 NOK and Herbert gets it for free since Herbert has access via his membership in the "Employee_NTNU"-group.

12.2.4.3 Example 3: Collaboration on Projects – Grand Scale

Users of the system may want to work together on documents. On a grand scale the job of adding free sheet music to the system is one collaboration project. As mentioned earlier, all users should be able to browse and download the free sheet music. To add and edit sheet music a user must be registered. This can easily be incorporated in to the system.

The reason users should be registered to edit sheet music is to handle the Change Notification and Document Restoring functions mentioned in 10.3.4.2 and 10.3.4.1.

Users that do not log in are automatically assigned to the Group "Default_group". This Group has read access to many public parts of the system. To add or edit a document a user must be logged in. All logged in users are assigned to the group "Logged_in". This Group has write access to the Collection called "Editable_sheet_music". This is illustrated below. A user that is logged in is a member of both "Default_group" and "Logged_in". Logged in users may then collaborate on the sheet music project since they have write access to that Collection.

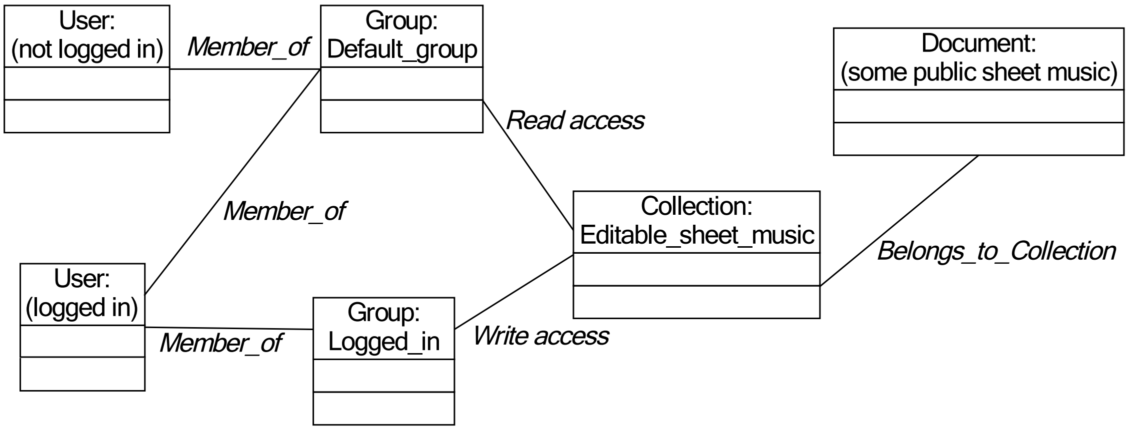

Figure 25: Diagram of access to the public collaborate sheet music collection. Logged in users may edit while other users may only read.

12.2.4.4 Example 4: Collaboration on Projects – Small Scale

A small scale collaboration project may involve only a few people working together on the same documents. Only those people are allowed to browse, read and write the documents. After the documents are finished, the access right may be changed by assigning the documents to other Collections.

Figure 26 shows how this can be done. Peter has made a document called "Collaborate_report", and is the owner of the document. He also makes a new Collection called "Collaboration_1", and link the document to that collection. He then makes a new group in the system called "Small_group_1". He makes a relation between "Small_group_1" and "Collaboration_1" with write access. All members of the Group "Small_group_1" may then edit the "Collaborate_report". Peter can then invite other users of the system to the Group "Small_group_1" which he is the owner of. He invites Herbert. They may now work together on the document.

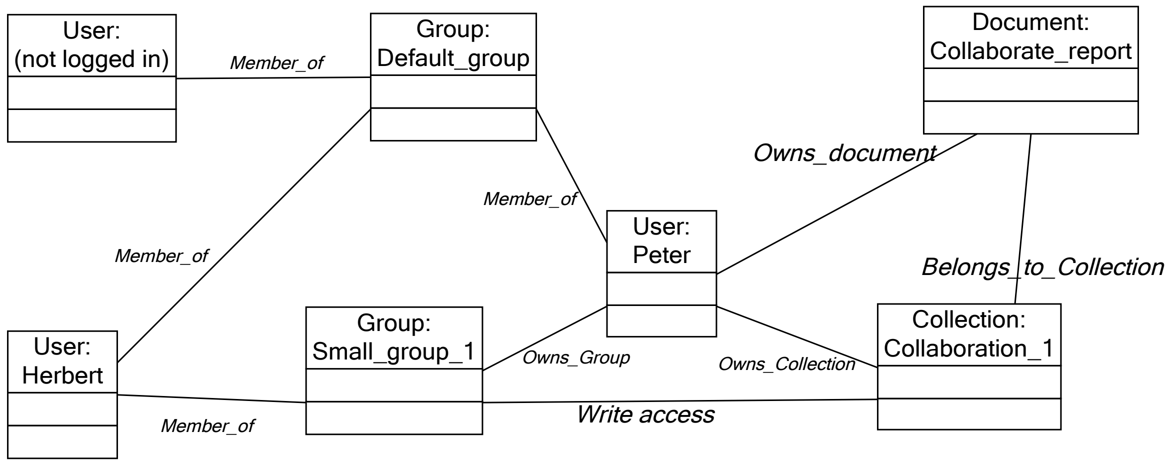

Figure 26: Diagram of a small scale collaboration project between Peter and Herbert.

Two new types of relationships are introduced here. The new relationships are the "Owns_Group" and the "Owns_Collection" relationships. The owner is the User that makes the Group or Collection. He may set the access rights and visibility of the Groups and Collections which he owns.

After a few months, Herbert and Peter decide that they want to make the "Collaborate_report" available for others to read. Peter then gives read access to the Group "Default_group". This is shown in the next figure:

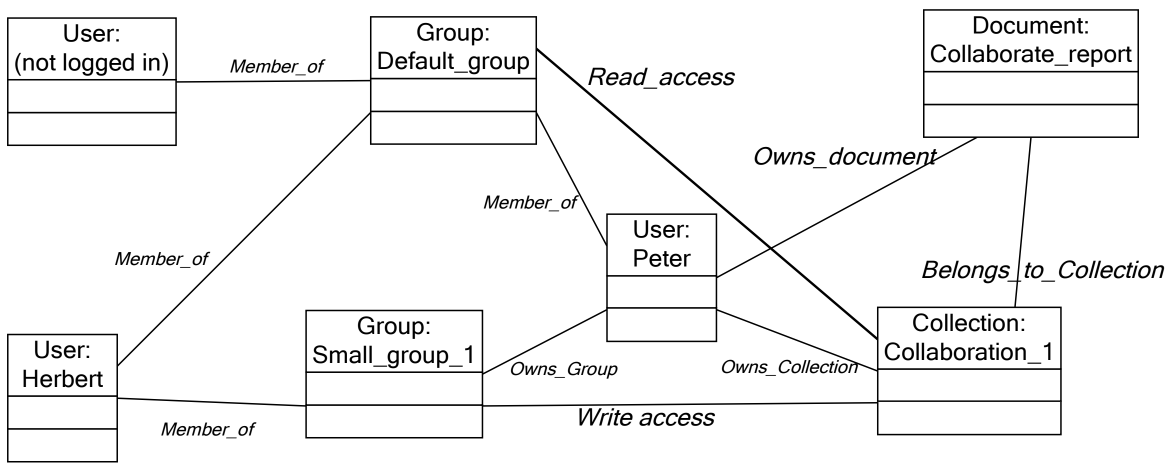

Figure 27: Diagram showing the small scale collaboration projects after it is made public. A new relation between "Default_group" and "Collaboration_1" is added which gives read access.

12.2.5 Comprehensive Access Model

A more comprehensive model of the access management is shown in the figure below. It expands the model in Figure 22 with three new relationships:

- Owns_Collection: A Collection has one owner, and a User may own many Collections. The User can decide which Groups shall have access to the Collection. Documents that he owns may be linked to the Collection.
- Owns_Group: A Group have one owner, and a User may own zero or more Groups. The owner of a Group decides who shall be members of the Group.
- Belongs_to_Document: One Document may consist of several other Documents. For instance the music book in Example 1, where the whole book is one document, and the particular melodies are documents themselves.

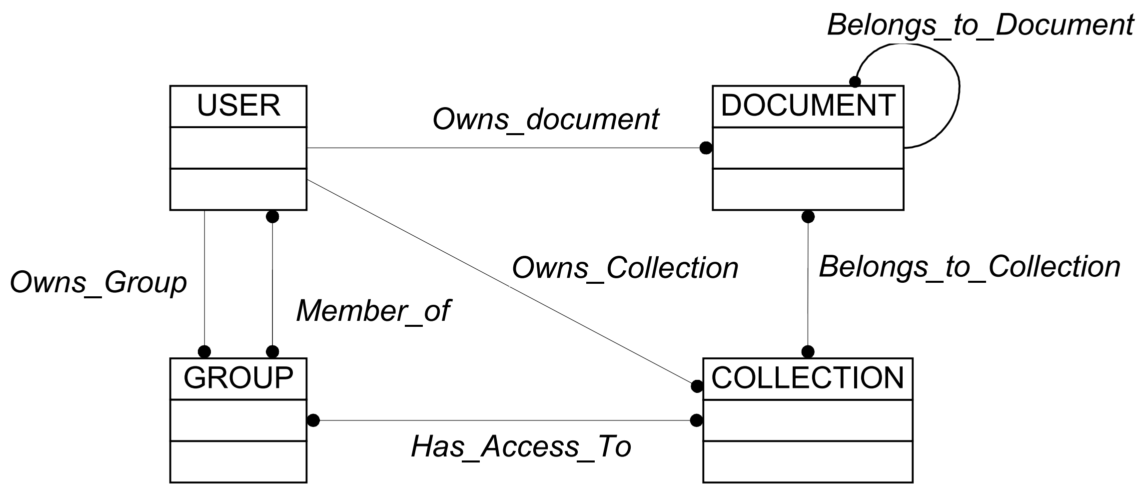

Figure 28: UML class diagram showing the relation between User, Group, Document and Collections.

12.2.5.1 User Access Management

So far we have not said much about how users are registered in the system. For this we suggest a solution with two types of users.

The first type of user (Type 1) is a user that may create both documents, collections and new groups. This type of user must register with the operators of the system, perhaps by manually signing a document which gives access. This is intended for users that belong to research institutions, universities and so on. These users may start collaboration projects together and use the system in their daily work.

The second type (Type 2) is a restricted user who is not allowed to make new groups or collections. This type of user is intended for the public collaborate sheet music project and similar projects. The registration of these users can be automatic via a website. These users do not need to be registered with their real identity, since the sheet music project should have no use for such information. It will also make the registration process easier when it is not necessary to check this information.

Another way to model this is to have only one user class in the database. Each user has a set of properties which tells what rights he has in the system. A person may register as a User automatically with a registration server. After registering with the system, a user get (or makes) a username and password. He is then a Type 2 user. When he signs a contract, the operator may change the settings for the user so he becomes a Type 1 user. This is seen in Figure 29. An operator assigns what rights the user has in the system.

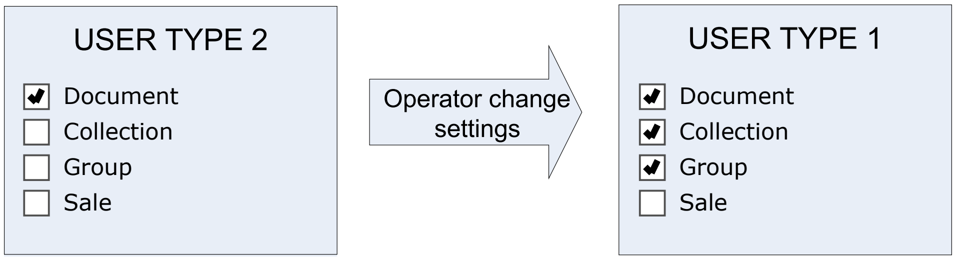

Figure 29: User management. When a Type 2 user signs a contract he may become a Type 1 user and get access to make Collections and Groups in the system. Operator are responsible for such changes.

A user that have not logged in is as mentioned interpreted by the system as the User “Anonymous”. This user simply has none of the checkboxes checked and consequently has no rights other than viewing public documents.

A fourth checkbox is also suggested. This is meant for Users who are allowed to sell Documents, which is a feature perhaps not everyone should have access to. A user which has the Sale checkbox checked is allowed to set prices for Documents in the Belongs_to_Collection relationship.

12.2.6 Summary

The database we developed in 12.2.4 and 12.2.5 will be able to handle many different scenarios. It is a combination of a closed and an open system. It allows some functions to be publicly available, while other functions are reserved for special users. The examples showed some of the supported scenarios and it can probably support many other scenarios as well. This design makes it possible to perform changes in uses access, documents and collections to suit a changing environment.

12.3 Use of Existing Systems

The problem assignment asks to consider existing systems for reuse. This may save a lot of work compared to having to make completely new systems. This section suggests possibilities for reuse in an implementation of the system.

12.3.1 Software

We suggest the following regarding the use of existing software:

12.3.1.1 Digital Library Software

Most of the functions in system can probably be implemented using the Greenstone 2 framework for digital libraries[48]. This framework has been used for many existing public library systems, including systems with audio, video and sheet music.

Greenstone is open-source, multiplatform software developed to be highly configurable. It supports many different user interfaces and document formats. Greenstone support structuring documents in collections which are easily maintained and accessible via the web. The system also supports distributed collections.

All operations on the Greenstone system is done via a web interface. To build collections using Greenstone a interface called the Collector is used. This enables the building of collections on remote machines. To build collections users have to log in using Greenstone's own security system. The security system prevents unauthorized users from building collections. Accounts for users can be set up to allow users to build collections or access private collections. To support more advanced features it is also possible to build and access collections using a Java interface[2].

12.3.1.2 Payment

The payment can be handled using the KOPEK[49] system. The KOPEK software is free to use, but there is a small cost for transactions. The KOPEK system can be used without doing changes to the existing system, and may therefore be a system that integrates easily with Greenstone.

In a big scale implementation such as the ENOMA project developing a special payment system may be cost effective.

12.3.1.3 Content Management

MediaWiki[50] can be used for some content management task. For instance collaborating on making a database of composers where users may contribute with information. However, it might be problematic to connect the MediaWiki with Greenstone since the who systems are

48 See 6.1
49 See 7.1.1
50 See 6.2.1

based on different technologies. MediaWiki may at least serve as an example collaboration system.

Storing information in two different systems means that the search may have to be integrated in an advanced way. The Greenstone 2 system also contains content management systems. Using Greenstone for content management may be the simplest solution because it reduces the number of systems that must be integrated.

A discussion forum is probably also straightforward to implement using the Greenstone system.

Using other systems such as phpBB may mean the same interoperability problems as with MediaWiki, although a plugin to handle phpBB data could probably be made.

12.3.2 Document Formats / File Formats

Chapter 3 gives an evaluation of the existing formats for storing audio, video and sheet music in a digital music archive. The most important results are listed here.

12.3.2.1 Audio

For the storage of audio we recommend following the IASA guidelines as stated in Chapter 3.1.2 :

Sampling Rate:	48kHz minimum, 96kHz preferred
Bit Depth:	24 bit minimum

Born-digital material should (at least) be stored at the same sampling rate and bit depth as it was created.

If compression is considered, only lossless compression should be used for storing the audio master files. Such compression can reduce the file size considerably (50-60%), and save both storage space and network transfer capacity without degrading the contents. We then recommend using a format such as FLAC[51] since it is free and seems to enjoy an increasing support in software. There is a catch with lossless compression though: If the compression and decompression algorithm is changed is may be difficult or impossible to retrieve the original signal. The safest approach is to use standard, uncompressed WAV(or BWV) files.

For presenting audio to the users the system should provide a choice of the uncompressed original format (or the file with lossless compression) and a compressed format such as MP3 or Ogg Vorbis. The target bitrate for the compressed files depends on the use of the file. For listening to music on portable devices like an MP3-player, a MP3 file at 128 kbit/sec may be adequate. For most people it is impossible to hear the difference of a 160 kbit/sec MP3 file

51 See 3.1.2.2

and a 44.1 kbit/sec WAVE file, but as mentioned earlier artifacts from the compression may be apparent if the MP3 file is to be further processed in audio software.

12.3.2.2 Video

We have not established a clear winner for the representation format for video storage. At least not for storage of video digitized from analogue formats. The main storage format should be uncompressed. NISO Framework Advisory Group recommend storing video on the media-dependent forms for now, or it can be stored on uncompressed or minimally compressed formats as D1, D5 or DigiBeta.

For presenting video to the user NISO[11] recommends:

1. High bandwidth LAN: MPEG-2 or MPEG-4 (high bit rate, high picture size)
2. Low bandwidth and web: MPEG-4, RealVideo or QuickTime (lower bitrate, lower picture size)

12.3.2.3 Sheet Music

Chapter 3.3 gives suggestions for a notation language for sheet music. There is no clearcut winner here, and both LilyPond and MusicXML may work. The discussion in section 3.3.6 compares the two formats for use in a digital music archive.

In addition to storing sheet music as a notation file the original scan should also be stored. It might be interesting to see how the score actually looked like, and this can probably best be done storing it as an high resolution image file.

12.4 Implementation Aspects

The system have not been implemented as part of this work. Some aspects of how it can be done are included here to function as a guideline.

12.4.1 Development of the Network of Archives

The intention is that each university or institution can run an individual archive system. This enables the institutions to have control of their users and documents. Later the individual systems can be interconnected in such a way that the public contents of them may be accessed from one web portal. This is accomplished by having standardized connection interfaces for each individual system that can be accessed from any other compatible archive system. This may make it easier to handle rights management since only the laws of the particular country applies to the system.

A central server for user names can be used to prevent the same user names to appear in different systems. This name server does not need to have all the user details, only a list of used names that the individual archive servers can check against.

12.5 Future Work

Future work on the digital archive for scores and music would include the following steps:

- Specify the unfinished part of the nonfunctional requirements
- Specification of metadata requirements
- Design of the document and user databases
- Design of a system for long term secure storage of data
- Implementation of the system using the specified systems
- Testing of the system

Part IV Conclusion

13 CONCLUSION

The work on the thesis has given me a great insight into the development of a digital music archive.

One of the main results in the thesis work is the specification and the resulting design. This has been based on the needs and requirements gathered during the thesis work.

For a digital music library to be useful it I have argued that good search mechanisms are needed. Several search methods have been described and suggested for the system. To enable effective search it is important to have an good representation of the material. Special emphasis has been put on finding a good representation language for sheet music. The representation should support both searching, printing, being displayed on screen and editing. LilyPond and MusicXML have been suggested for this purpose. For storing data for preservation purposes the data should be stored so that as much as possible of the original information is retrievable. This has guided the suggestions for storage of audio and video data.

In the design chapter I argue that the system should be a mixture of an open and a closed system regarding user access. The freedom of open access to for instance a public database of free sheet music can be supported in a system. At the same time some users may share non-public documents with each other. An user access database to support this has been developed and is described in the design chapter. Further, suggestions for reuse of existing software to implement the system is given.

When the system is implemented is should be possible to support many different use scenarios. These include:

- Publication of research material with musical examples both as audio and sheet music
- Research in the music material contained in the archive using different search methods
- A record company function where institutions can sell music or other kinds of documents
- Public access to a sheet music database that everybody can contribute to
- Co-operation on research projects for groups of users
- Interconnect music archives located at geographically different places.

Part V Reference Section

14 REFERENCES

[1] *Fiol Product Description*. Last visited: May 24, 2006. http://www.fiolmedia.com/files/Produktblad_FIOL_en.pdf

[2] I.H. Witten, D. Bainbridge. *How to build a Digital Library*. Morgan Kaufmann, Amsterdam, 2003

[3] IASA-TC 03. *The Safeguarding of the Audio Heritage: Ethichs, Principles and Preservation Strategy*. IASA, September 2001. Last visited: May 6, 2006. http://www.iasa-web.org/iasa0013.htm

[4] IASA-TC 04. *Guidelines on the Production and Preservation of Digital Audio Objects*. IASA Technical Comittee, 2004

[5] *Lecture notes in TTT4190 Musikkteknologi*, NTNU, Spring 2005

[6] Robin Whittle. *Lossless audio compression*. Last visited: June 14, 2006. http://www.firstpr.com.au/audiocomp/lossless/

[7] *FLAC - Free Lossless Audio Codec*. Last visited: April 20, 2006. http://flac.sourceforge.net/

[8] Doug Tougaw. *Lecture #39: MP3 Audio Compression*. Last visited: May 13, 2005.

[9] B.G. Haskell, A. Puri and A.N. Netravali. *Digital Video: An introduction to MPEG-2*. p. 89. Chapman & Hall, Ltd., London, UK., 1996

[10] F. Pavuza. "Format and File Issues for Video Archiving". In *seminar on "file formats for preservation" (Electronic Resource And Preservation Network)*, Austrian National Library, Vienna, Austria, 2004. http://www.erpanet.org/events/2004/vienna/presentations/erpaTrainingVienna_Pavuza.pdf

[11] NISO Framework Advisory Group. *A Framework of Guidance for Building Good Digital Collections. 2nd edition*. Bethesda, MD: National Information Standards Organization, 2004. Available from: http://www.niso.org/framework/framework2.html

[12] *Wikipedia: MXF*. Last visited: May 27, 2006. http://en.wikipedia.org/wiki/MXF

[13] *Wikipedia: Container format*. Last visited: May 26, 2006. http://en.wikipedia.org/wiki/Container_format

[14] *Wikipedia: MPEG-2*. Last visited: May 25, 2006. http://en.wikipedia.org/wiki/MPEG-2

[15] A. Lie. "MPEG-4, audio, visual". In *lecture slides in TTM4140 Videoteknologi*, Dept. of Telematics, NTNU, 2002. http://www.item.ntnu.no/fag/ttm4140/host04/MPEG4/MPEG-4%20Audio%20and%20Video%20II.pdf

[16] *SMPTE.org: SMPTE Releases VC-1 Standard (Press Release)*. Last visited: May 26, 2006. http://www.smpte.org/news/press_releases/003_06.cfm

[17] *Wikipedia: Windows Media Video*. Last visited: May 25, 2006. http://en.wikipedia.org/wiki/Windows_Media_Video

[18] *Wikipedia: QuickTime*. Last visited: May 25, 2006. http://en.wikipedia.org/wiki/Quicktime

[19] *Wikipedia: RealVideo*. Last visited: May 26, 2006. http://en.wikipedia.org/wiki/RealVideo
[20] *RealNetworks.com: Realvideo*. Last visited: May 27, 2006. http://www.realnetworks.com/products/codecs/realvideo.html
[21] *Wikipedia: AVI*. Last visited: . http://en.wikipedia.org/wiki/AVI
[22] J. Riley and J. Dunn. "Digitizing and Delivering Audio and Video". In *workshop slides*, Digital Library Brown Bag Series, Feb, 2005. http://www.dlib.indiana.edu/workshops/bbspring05slides/digitize/bbFeb2005.pdf
[23] J. Riley and I. Fujinaga. "Recommended Best Practices for Digital Image Capture of Musical Scores". In *OCLC Systems & Services*, 19, no. 2, pages 62-69, June 2003. http://www.dlib.indiana.edu/~jenlrile/oclc/oss.pdf
[24] *Sibelius Scorch*. Last visited: May 2, 2006. http://www.sibelius.com/products/scorch/index.html
[25] *MusicRAIN*. Last visited: May 5, 2006. http://mediarain.com/musicrain/
[26] *MIDI 1.0 Specification*. Sequential Circuits, Inc., San Jose, California., Aug 5, 1983
[27] Castan, Gerd*Musical notation codes*. Last visited: May 7, 2006. http://www.music-notation.info/en/compmus/notationformats.html
[28] E. Selfridge-Field. *Beyond MIDI: The Handbook of Musical Codes*. MIT Press, Cambridge, Massachusetts, US, 1997
[29] *GUIDO Noteserver*. Last visited: May 7, 2006. http://www.noteserver.org/noteserver.html
[30] *The SALIERI Project: GUIDO homepage*. Last visited: May 7, 2006. http://www.salieri.org/guido/
[31] Choudhury et al.. *Strike Up the Score - Deriving Searchable and Playable Digital Formats from Sheet Music*. D-Lib Magazine, Vol. 7, No. 2, Feb 2001. http://dlib.anu.edu.au/dlib/february01/choudhury/02choudhury.html
[32] *LilyPond Introduction - LilyPond in Practice*. Last visited: May 7, 2006. http://sca.uwaterloo.ca/lilypond/lilypond.org/web/switch/tour
[33] *Orpheus Music Search*. Last visited: May 7, 2006. http://teuge.labs.cs.uu.nl/Rntt/mir/mir.cgi?pwd=none
[34] *Lilypond Manual*. Last visited: May 7, 2006. http://wikitex.org/doc/lilypond.pdf
[35] *Lilypond F.A.Q*. Last visited: May 9, 2006. http://lilypond.org/web/about/faq
[36] *Recordare.com: MusicXML Software*. Last visited: April 18, 2006. http://www.recordare.com/software.html
[37] *Metawiki: Music Markup*. Last visited: April 13, 2006. http://meta.wikimedia.org/wiki/Music_markup
[38] J. Beggs and D. Thede. *Designing Web Audio - RealAudio, MP3, Flash and Beatnik*. O'Reilly & Associates, Inc., CA, US, 2001
[39] G.C. Demetriades. *Streaming Media - Building and Implementing a Complete Streaming System*. Wiley Publishing, Inc., Indiana, US, 2003
[40] M. Dawe and B. Finn. *PhotoScore 4 Professional - User Guide*. 16th edition, Neuratron Limited, 2005
[41] Choudhury et al. *Document Recognition for a Million Books*. D-Lib Magazine, Vol. 12, No. 3, March 2006. http://purl.pt/302/1/dlib/march06/choudhury/03choudhury.html
[42] *The Gamera Project Homepage: A gentle overview of Gamera*. Last visited: May 31, 2006. http://dkc.jhu.edu/gamera/html/overview.html

[43] *Gamera download page*. Last visited: May 31, 2006. http://linux.softpedia.com/get/Science-and-Engineering/Image-Recognition/Gamera-2103.shtml

[44] M. Droettboom, I. Fujinaga et. al. "Using the Gamera framework for the recognition of cultural heritage materials". In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, Building and using cultural digital libraries, pages 11-17, Portland, Oregon, USA, 2002.

[45] I. Fujinaga and J. Riley. "Digital Image Capture of Musical Scores". In *Proceedings of the third International Conference on Music Retrieval*, Paris, France, October 13-17, 2002. http://ismir2002.ismir.net/proceedings/03-SP01-3.pdf

[46] *Filmarkivet.no*. Last visited: April 23. http://filmarkivet.no/

[47] NSA Presentation Folder, 2006

[48] Announcement text for NRK Programarkivet, spring 2006

[49] *Norwegian Institute of Recorded Sound*. Last visited: May 2, 2006. http://www.recordedsound.no/

[50] Email from Norsk Lydinstitutt, June 1, 2006

[51] D. Bainbridge, K.J. Don, G.R. Buchanan, I.H. Witten, S. Jones, M. Jones and M.I. Barr. "Dynamic digital library construction and configuration". In *proceedings of the European Digital Library Conference on Digital Libraries, ECDL 2004*, Bath, England, 2004. http://www.sadl.uleth.ca/greenstone3/Dynamic_DL_Construction.pdf

[52] *Greenstone Digital Library System*. Last visited: June 4, 2006. http://www.greenstone.org/cgi-bin/library

[53] G. Buchanan, D. Bainbridge, K.J. Don and I.H. Witten. "A New Frameworkfor Building Digital Library Collections". In *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 23 - 31, Denver, CO, USA, 2005.

[54] *Wikipedia: Comparison of Content Managment Systems*. Last visited: May 5, 2006. http://en.wikipedia.org/wiki/Comparison_of_content_management_systems

[55] *Wikipedia: Content Managment Systems*. Last visited: June 1, 2006. http://en.wikipedia.org/wiki/Content_management_system

[56] *Wikipedia: Wikitex*. Last visited: May 31, 2006. http://en.wikipedia.org/wiki/Wikitex

[57] *Kopek Website*. Last visited: March 10, 2006 (Changed June 2006...). http://kopek.net/merchants_no.html

[58] Kopek Internet Payment System, information whitepaper by Kopek AS

[59] Kopek Internet Payment System, information folder by Kopek AS

[60] *Payex.no - Felles standard for betaling på nett*. Last visited: June 3, 2006. http://www.payex.no/no/merchant/paymethods.htm

[61] *Apple iTunes - iTunes - Download iTunes*. Last visited: June 4, 2006. http://www.apple.com/itunes/download/

[62] *The Register: iTunes DRM cracked wide open for GNU_Linux*. Last visited: June 4, 2006. http://www.theregister.co.uk/2004/01/05/itunes_drm_cracked_wide_open/

[63] *Hymn Project: Hymn Manual*. Last visited: June 4, 2006. http://hymn-project.org/docs/hymn-manual.html

[64] *VB NET Tutorials: Web Services - SOAP, WDSL, Disco and UDDI*. Last visited: June 2, 2006. http://www.exforsys.com/content/view/1560/350/

[65] Bainbridge et al.. "Visual Collaging of Music in a Digital Library". In *Proceedings of the 5th International Conference on Music Information Retrieval*, Barcelona, Spain,

October 10-14, 2004. http://ismir2004.ismir.net/proceedings/p072-page-397-paper223.pdf

[66] R. Typke, F. Wiering and R. Veltkamp. "A Survey of Music Information Retrieval Systems". In *Proceedings of the 6th International Conference on*, pages 153-160, London, UK, 2005.

[67] J. Conallen. *Building Web Applications with UML*. Pearson Education, Inc., Boston, US, 2nd edition, 2003

[68] I. Sommerville. *Software Engeneering, 6th edition*. Pearson Education Limited, Essex, England, 2001

[69] Schulzrinne et. al. *RFC 3550 "RTP: A Transport Protocol for Real-Time Applications"*, 2003

[70] Allman et. al. *RFC 2581 "TCP Congestion Control"*, 1999

[71] H. Schulzrinne et. al. *RFC 2326: " Real Time Streaming Protocol (RTSP)"*, 1998

[72] L.A. Rønningen. *lecture foils in TTM4140 Videoteknologi on Streaming, RTP, RTSP*, Oct 2003

[73] *Duke University - Sheet Music Search*. Last visited: May 10, 2006. http://www.lib.duke.edu/texis/smi/search/main.html?advanced=yes

Appendix A: Metadata

For sheet music the following fields are used for metadata by the Duke University Library service[73]. This list serves as an example of one possible way to standardize metadata. The list may also serve as a guide to what search fields should be included.

NAMES:
- Composer:
- Lyricist:
- Arranger:
- Performer:
- Book:
- Producer:
- Staging:
- Choreographer:
- Illustrator:
- Artist:
- Engraver:
- Lithographer:
- Publisher:
- Distributor:
- Movie:

TITLES:
- Title:
- First line:
- Refrain:
- Series:

SUBJECTS:
- Graphics:
- AAT:
- LCSH[52]:
- Dedicatee:

PUBLICATION INFORMATION:
- Date:
- City:
- Country:
- Edition:
- Pagination:
- Plate no.:
- Publisher's no.:
- Form of composition:
- Score:
- Instrumentation:

52 Library of Congress Subject Heading

Recordings available:
HOLDINGS INFORMATION:
Collection/Call no./Copies:
Notes:

Appendix B: Problem Statement in Norwegian

Elektronisk arkiv for lyd og noter

Hensikten med arkivet er å gi mulighet for å publisere ukjent, ikke-innspilt musikk (lyd, noter, etc, ev. video) slik at den blir tilgjengelig for dagens utøvere og forskere. En sentral server med tilgangskontroll og brukergrupper skal inneholde et brukervennlig bibliotek. Tilgangskontrollen kan også omfatte betaling for tjenester. Opptak og tidligere utgitte CD-er kan legges inn.

Arbeidsoppgaver:

- Oppgaven består i systemintegrasjon
- Avklare behov. Distribusjonskanal for hva og for hvem (intervju av sentrale aktører/personer)
- Evaluere eksisterende systemer mht gjenbruk/påbygging
- Finne tekniske løsninger til rettighetsproblematikken ved nedlasting og bruk av lyd og noter fra nettverket. Foreslå måter å beskytte opphavsrett og betaling.
- Teste digitalisering av noter med OCR-programvare
- Spesifikasjon (eventuelt påbygging)
- Design (eventuelt påbygging)
- Gi en vurdering av hvordan systemet kan implementeres