

# MuTaTeD'II: A SYSTEM FOR MUSIC INFORMATION RETRIEVAL OF ENCODED MUSIC

Carola Boehm, Dept. of Music, University of Glasgow  
Donald MacLellan, Dept. of Music, University of Glasgow  
Cordy Hall, Dept. of Computing Science, University of Glasgow

## ABSTRACT

The proof of concept project MuTaTeD! validated the concept of integrating two existing music representation standards: SMDL (Standard Music Description Language) and NIFF (Notation Interchange File Format). An application with the combined implementation of these two standards, SMDL and NIFF, supports the representation of music as a time-structured entity in its own "gestalt" as it is conceived and yet provides a standard for high-quality display via the NIFF format. One of the most important results coming out of this first project was the first SMDL to NIFF converter, allowing NIFF compliant Notation Editors to display SMDL encoded music. Building on the results of this project, MuTaTeD'II has started in November 1999. Its aim is to design and implement a music information retrieval system with delivery/access services for encoded music. The prototype service will provide a user friendly, web-based search/browse/query interface to access musical content.

## INTRODUCTION

Objectives of the project MuTaTeD were: to integrate SMDL - ISO/IEC DIS 10743 (Newcomb, 1991) as the Model with NIFF (NIFF SPEC, 1995) as one possible View, and establish a standard Meta-DTD for music tagging languages, which could be used by the wide user community. Additionally, it was to research into the development and integration of a SMDL DTD for the wider music user community. The work heavily influenced the "Structured Music MPEG7 proposals" (Boehm/Hall, 1999) which were proposed in order to ensure an SMDL-compliant standard.

Due to the time and financial constraints of this project an early decision was taken to use freely available software development environments. For the parser and compiler the Lex/Yacc Parser Technology (lexer, parser, code generator) was chosen and a multipass compiler has been developed with this technology. To create the NIFF output files the NIFF Software Development Kit, also freely available, was used. In addition to this, we could utilise already existing multipass compilers, which convert SMDL to NIFF, developed by the CANTATE project (developer: Steven Mounce) (CANTATE 5-3, 1997)

Although proving the initial concept by having accomplished a prototype which was able to convert SMDL to NIFF for a restricted set of music examples (music upto 7 sharps and flats in time signatures of 4/4), we were aware of the restrictions imposed on this prototype due to the choice of technologies to be used and the time restraints on the whole project. The DTDs had to be hardcoded into the parsers, which meant that future use of these parsers with slightly different types of music or DTDs was a matter of adding or changing the code. The second pass of the compiler restricts handling to a single line using treble clef, where as the first pass is able to process other clefs, and upto five staves. Although it

would have been easy to build a second project on top of the same technology, expanding the parsers and converters to a much more fuller scope existing in traditional music, we wanted to base the follow-up project on a technology which would be able to tackle these issues without the restrictions of the former project. To utilise the full portability and the scalability of the original SMDL standard we chose to use the Groveminder system from TechnoTeacher Inc. as the basis for the MuTaTeD'II System. Using Grove technology (Newcomb, 1999) and its underlying object-oriented database management to store the groves, as described later in the article, we started to build a new system with, as we hope, a bigger promise for the music information retrieval community of the future. This paper gives an overview over both projects. More details of the two projects can be found at their relevant website. (MuTaTeD1 WWW 1999, MuTaTeD2 WWW 2000)

## MuTaTeD!1 - THE TECHNICAL SET-UP

Already in MuTaTeD!1 we saw that the ideal system setup would be to distribute the multipass converter across server and client whilst using a platform independent client.

This design<sup>1</sup> (see Fig.1) would have had the advantage of client-server distribution in order to maximise the performance for the user. The conversion processes to the final binary NIFF files would be executed on the clients' machines, thus minimising the downloading time and platform dependencies. Only text-based information would be sent to the client, the binary being created within the client-side application. Additionally this would

---

<sup>1</sup> For future notation within this document, a file written in SGML and using an X DTD is described as being in SGML(X). A file written in C and using functions from the NIFF SDK is referred to as being in C(NIFF SDK).

have the advantage of involving separate passes, allowing the modularity to add different client side converters, thus possibly adding other tag-based music description languages, such as for instance GUIDO (Hoos/Kilian/Hame/Renz, 1998).

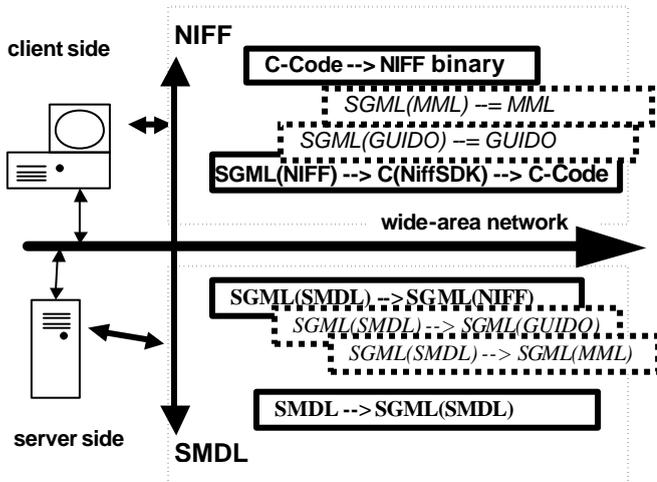


Fig. 1

Facing the problem that the NIFF SDK was available only for Unix, and not having the time for tedious recompilation procedures of the software tools for WinNT, the decision was made to put all of the passes onto the server side (see Fig. 2).

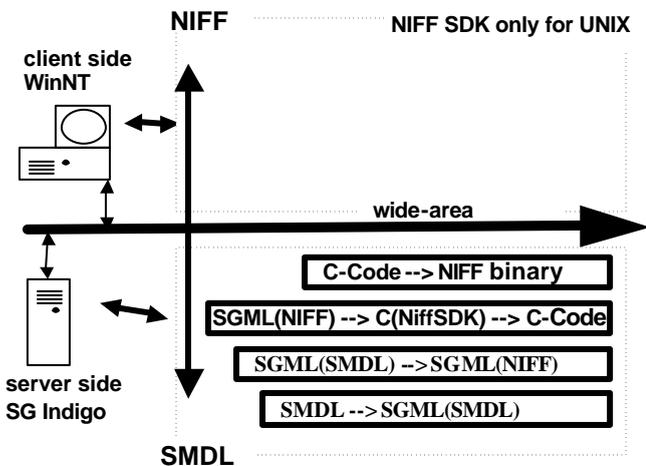


Fig. 2

### THE MuTaTeD! MULTIPASS COMPILERS

As any compiler our programs contained a *lexer*, a *parser* and a *code generator*. The lexer splits up the program text into what are called 'tokens', tiny pieces of information. These are then examined by the parser, which builds a tree representing the program's structure. If the program can be parsed successfully, then the tree is well-defined and is passed on to the code generator, which walks over the tree and produces target code as it examines each node in the tree.

The compiler developed in MuTaTeD! worked with two passes. The first translated SMDL into an intermediate form of NIFF, called SGML(NIFF). The second translated into NIFF binary code. Each of the two passes was written using LEX and YACC. The second pass translated an SGML(NIFF) file to a C program, using the NIFF SDK. This was compiled and executed, resulting in a NIFF binary file that can then be loaded into a NIFF compatible music editor, such as LIME. We have made these compilers for the MuTaTeD! project freely available for the developer's community, and hope that interest exists to use this technology to expand upon.

Having had a small, but existent influence into MPEG7 as a side product of the project, we hope to be able to continue and expand this influence of the MPEG7 standardisation process to include compatibility with SMDL. Newest important developments within the finalisation of the SMDL standard will include compatibility with XML and HyTime2, making this standard even more interesting to work with.

### FROM MuTaTeD! TO MuTaTeD'II

Besides proving the concept of using these two standards, the use of the LEX/YACC technology enabled us to implement the converters, but also restricted our flexibility to certain extent. With LEX/YACC we had to hardcode our chosen DTD's into the converters. This restriction was the major reason to implement the follow-up system in the project MuTaTeD'II differently, as we had less of a time and resource constraint. The MuTaTeD'II team decided to use the Groveminder Technology, which, besides other advantages, will allow for the reading-in of different DTDs. NIFF, being binary, also posed some problems of accessing and searching the content. The fact that any compilation/search procedure has to always be preceded by a reading in or out of the whole binary data in the case of binary formats, has consequences in the design of intuitive content music retrieval, specifically for displaying dynamically parts of music. So although NIFF has proven to be the powerful format, which it is, we are contemplating the use using other text based representation language for future implementations. As depicted in Fig. 1 the effort involved in implementing compliance with other tag based description languages is not immense.

### MuTaTeD'II, THE TECHNICAL SET-UP

In MuTaTeD'II, aiming to design and implement a working examples of a music information retrieval system with delivery/access services for encoded music, the same basic system architecture was being set up, this time using Groveminder's support for tagged based languages and its' needed parsers and compilers. The prototype service will provide a user friendly, expandable web-based search/browse/query interface to access musical content.

Both high level searches and more complex low level searches are hoped to be catered for. High level searches will include searches for specific metadata regarding composer or date of composition. As SMDL supports the storage of a certain set of bibliographic metadata, this will be utilised within this system. Low level searches will be able to concentrate on musical aspects within each score or across a pre-selected collection of scores. This low level searching would give the user the ability to search for a certain pitch, note or articulation patterns. In the next version of the system it is hoped to be able to provide any combination of pitch, note and articulation searches together, other SMDL elements, such as lyrics or similar, are possible. This would seamlessly transform a search activity into a tool for music analysis.

Two main API's are used, both written in C++. The first is the Groveminder system itself which is used to lex, parse and search SMDL files. The second is GNU Cgicc. This is a gnu open source API which provides functions for talking to a HTTP server and also creating HTML on the fly. As described below in detail, the system internal steps involved in the successful indexing for the searching process include:

1. Opening and parsing a SMDL document for validation, thus building the grove.
2. Parsing the resulting tree/grove to locate the information searched for.
3. Create containers and store the data in the containers.
4. Apply the search functions to the containers and return the results.

The Groveminder technology allows the construction of groves from any valid SGML document. A document is opened, parsed and validated by Groveminder. A SGML grove is then constructed ready for the second stage of processing.(See Fig. 3)

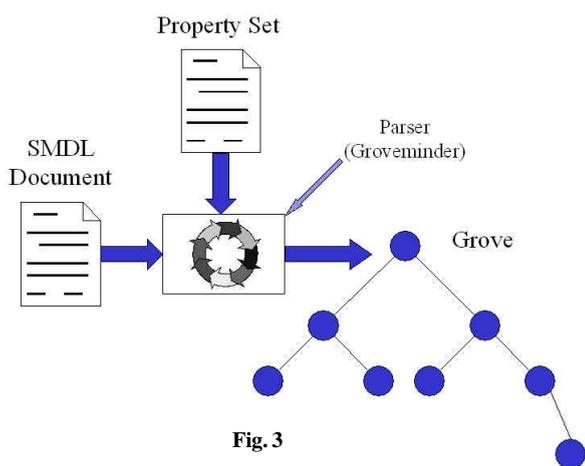


Fig. 3

Groves are part of the HyTime standard (an extension of SGML) and provide a way for systematically describing the in memory structure of a SGML file. They can best be described as a concrete data model which provides a

predictable data structure for the data. Groves are constructed by parsing a SGML document and applying an object model called a property set.

The property set is unique for each different type of grove you want to construct but has a close relationship with the DTD of the source document. The DTD for a SGML file contains elements that have attributes and sub-elements which in turn have attributes and so on. This is the basis of providing hierarchical structures in SGML. Similarly a property set contains classes that have properties and sub-classes. (See Fig. 4)

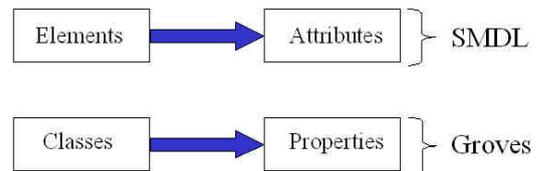


Fig. 4

In addition to the parsing processes supported by Groveminder, it also contains functions for navigating the grove and locating data. These functions are used to isolate data specific to the current search being carried out. As the data is found it is streamed into containers that will enable the data to be searched.

A diagram of the basic structure of the containers can be seen in Fig. 5. This shows the musical information required to do a variety of searches has been isolated with all other extraneous information discarded.

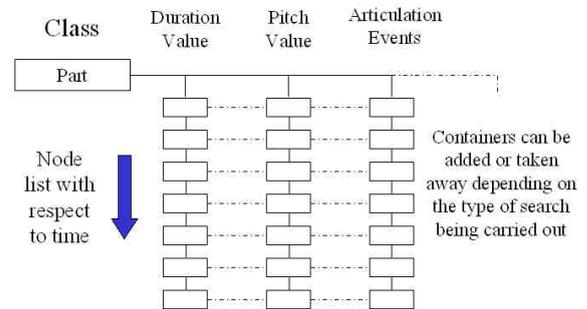


Fig. 5

After the necessary searching has been carried out the results are returned via the Cgicc API which creates the search results web page on the fly.

Optionally using pointers instead of copying the data across to containers would result in inefficient searches due to the high number of cache misses when searching through the data. In addition to this, it might be needed to free up memory through destruction of groves, which would not be possible with using pointers instead of the copy-to-container method.

Essentially the method of creating new in-memory data models containing only search specific information, promises a very efficient and very expandable basic software design. It allows the non-proprietary ethos of

SGML to be extended to the way data is constructed and referenced to. Although this conformity to a data model doesn't dispense with the need for specific code to be written for SGML files with different DTD's it does provide a number of advantages.

1. Using Groveminder automatically gives us the support and the existence of a standard data model, which is closely linked to our source data.
2. By using a standard object model like groves and property sets the interoperability of a wide variety of different applications is easily enabled. (For more information on Groves see (The XML Cover Pages))
3. Portability across platforms.

This is of immense benefit when expanding the system in light of future developments.

### CONCLUSION

In the past the availability of music manuscripts on the net has been hampered for a variety of reasons. One of the main ones being the propriety nature of the file formats used in most score notation packages. Throughout the Mutated projects we have tried to overcome some of these restrictions. In MuTaTeD! the task to make SMDL files viewable was undertaken by creating SMDL to NIFF converters. This opened the door for the next obvious step which was carried out in MuTaTeDII. SMDL being a tag-based language was one of the obvious choices for transporting music information over the internet.

While the process of creating a fully interactive online music information search and analysis tool is still in its embryonic stage it is immediately obvious that a solution of this nature could have a variety of benefits to the music community.

1. high-level to low-level music search facilities
2. adaptable, expandable analysis tools
3. use of an underlying powerful, standardized description language, which does not necessarily restrict the handling of purely encoded music
4. expansion possibilities into areas of audio/video tagging

Finally the expansion of the web and the general increase in tag based languages like XML and SGML mean that in the future there will be a greater provision for developing applications that deal in these languages. The next generation of browsers (i.e. Netscape 6) will be fully XML compliant and given time SGML (or something of a similar level of complexity) will be more commonly used. This drive towards easier interchange of information is inevitable and provides an opportunity to put data back in the control of users rather than leaving them prone to the fickle trends of commercial application developers who

more often than not have been responsible for the bottleneck in information interchange.

### REFERENCES

- [Boehm/Hall 1999] Boehm, Carola Boehm and Hall, Cordy. 1999. "ISO/IEC JTC1/SC29/WG11/ MPEG 98/P620/W2463, MPEG proposal: Description Scheme for description of music content", Vancouver: MPEG 1999.
- [CANTATE 5-3, 1997] "Cantate, Download Deliverable 5-3 : Development Model with Summary and Recommendation", Amsterdam 1997.  
[http://www.svb.nl/project/cantate/cant\\_deliv.htm](http://www.svb.nl/project/cantate/cant_deliv.htm) 2000-06-02.
- [Hoos/Kilian/Hame/Renz, 1998] Hoos, K. A. Hamel, K. Renz, J. Kilian. 1998. The GUIDO Music Notation Format - A Novel Approach for Adequately Representing Score-level Music (H. H. published in: ICMC'98 Proceedings, p.451-454) Specifications at <http://www.informatik.tu-darmstadt.de/AFS/GUIDO/21/02/00>
- [Newcomb, 1991] Newcomb, Steven R. "Standards. Standard Music Description Language Complies with Hypermedia Standard." IEEE Computer 24/7 (July 1991) p.76-79. ISSN: 0018-9162.
- [Newcomb, 1999] Newcomb, Steve. 1999. Technical Description, <http://www.tecno.com/> 2000-06-05.
- [NIFF SPEC, 1995] Cindy Grande and members of the NIFF project team, Specification for the Notation Interchange File Format, 1995.  
<http://esi24.ESI.UMontreal.CA:80/~belkina/N/NIFF6a3.txt> 2000-06-05.
- [MuTaTeD1 WWW 1999] Hall, Cordy and Boehm, Carola. Website of the Project "Music Tagging Type Definition", MuTaTeD!1.  
<http://www.pads.ahds.ac.uk/mutated> 2000-06-05.
- [MuTaTeD2 WWW 2000] Boehm, Carola and MacLellan, Donald. Website of the Project "MuTaTeDII, A system for Music Information Retrieval of Encoded Music"  
<http://www.pads.ahds.ac.uk/mutated2.html> 2000-06-05.
- [The XML Cover Pages] Cover, Robin  
The XML Cover Pages website. Resource of everything to do with SGML and XML, including SMDL, groves and property sets  
<http://www.oasis-open.org/cover/> 2000-06-05.