Computer Based Music Notations as Object

[1]Dr. Mohammed Shafi, [2]Dr. D.K. Pandey,

[1]*Faculty of Computer Science, S.G.College, Gavimath Campus, Koppal-583231. Karnataka, India.*
[2]*Faculty of Computer Science, Magadh University, Bodh-Gaya, Bihar, India.*

Abstract:

The purpose of this study is to analyse the Western music notation system as a set of objects that act as computational representatives for notation symbols. The primary analytical method is the object-oriented analysis used in computer science. To improve coherency in the resulting analysis, I have extended standard object-oriented methodology with linear logic, a method also originating from computer science. The result of the study is a structural object model that describes the roles and relationships of objects found in music notation. This model can be used as a basis for computer software design.

The intention of the study is to represent a theoretical description of music notation that can be used in the design process of a music notation computer program. Previous studies on the computer-based music notation have shown that the processing of music notation involves different types of information. Existing musical data representations serve as current examples of how the various types of information can be encoded and of the kinds of difficulties involved. This study is based on the assumption that a consistent object model can formed by including only those objects that have an explicit visual appearance; information belonging to levels or domains other than the visual one can be included as properties of visual objects.

Keywords : *Music Notation System, Object Oriented Analysis, Music Notation*

I. COMPUTER BASED MUSIC NOTATION

Music notation presents a challenge for software developers for many reasons. First, music notation programs typically have to process several different types of information i.e., graphical, logical, performance and analytic. Secondly, the rule set of music notation is extremely complex compared to, for instance, word processing or other commonly-known computer applications. Third, the market for commercial music notation application is considerably smaller than that of, say word processing program. These facts mean that the development process of a notation program is typically difficult and slow, while the potential outcome of a commercial endeavour may be small.

The Computational and representational difficulties concerning music notation have also been subject to scientific research in both musicology and computer science. There has been active scientific research on many areas concerning music notation. These include user interfaces, forms and means of data input, different forms of displaying music notation, automatic spacing algorithms, algorithms for generating a musical performance from a notated score, optical recognition of printed and handwritten music notation etc., One central area of research is the computer representation of music notation, which also lies within the scope of this study.

Computer representations have been developed for various purposes related to music notation. These include data input, data output, the internal memory structures of notation programs, file formats for data storage, and file formats for data interchange between notation programs. This model comprises of hierarchically organized objects, can be used as a basis for designing concrete representations.

II. RESEARCH METHODS

Computer Science offers systematic methods for various tasks and stages in software engineering. These tasks include the selection and use of programming languages and the preparation of specifications for programming tasks. In addition, formal methods have been presented for analysing a system or for a problem to be implemented in or solved by computer software.

Object oriented analysis is a part of object-oriented software engineering methodology. Object oriented software engineering uses objects as a basic unit of software construction. An object is a combination of data and a set of operations for manipulating the data. The use of objects provides a formal method of decomposing a large and complex system into smaller parts that can be written and tested separately. An object can hide its internal data structures form other objects so that the data is protected from undesired use. Thus, the object's data structures may also be redesigned and changed without requiring changes to other objects.

Object oriented analysis is a method whereby a system to be implemented in software is decomposed into objects, and the basic relationships of these objects are defined. The result of the analysis is presented as graphical diagrams and textual descriptions.

Several formal object oriented methods have been presented. In the late 1990's, the Unified Modelling Language (UML) became the dominant formal object oriented methods. Thus UML notation was chosen for use in this study. However, UML is primary a modelling language, not a formal method of performing an analysis. To systematize further the analysis process, linear logic was used as a complementary method.

Linear logic is an alternative logic to classical logic. Linear logic is based on the principle of limited resources as opposed to the principle of unlimited resources in classical logic. For software engineering linear logic offers advantages for simplified management of memory and other computing resources. Linear logic provides a systematic method for evaluating various analytical decisions.

III. APPLICATION OF METHODOLOGY AND GENERAL GOALS

The basic goal of this study is to present a model of the Western Music Notation System, such that this model, is turn, can be used as a basis for computer software design. The model should be general enough to be applicable to many types of computer applications requiring music notations. At the same time, the model should be independent enough not to require a particular area of application. Also, it should be independent of the computer hardware or software environment and of the implementation programming language.

The original impetus for the present study arose from a technical interest in applying object oriented techniques to the design of music notation software. Previous personal experience with music representations and on music composition software design had shown that modularization can lead to an effective and efficient musical computing environment while requiring a relatively small amount of programming code. Then came further experience in designing and implementing an audio signal processing system. This experience showed that an object oriented approach provided that means of building an easily, maintainable and expandable framework, one that could also be accomplished with a relatively modest amount of programming effort. This leads to experiments with applying similar techniques to the design of music notation software. However, music notation proved to be a much more difficult problem than the one faced in signal processing or the manipulation of musical events. Experiments with different program prototypes showed that a theoretical study, separated from the practical problems of programming, was needed in order to form an objective view of the problem domain.

The present study focuses more on data representational aspects of computer-based music notation, and less on algorithms involved with musical data processing. More specifically, my focus is on high-level structural abstractions rather than on low-level data structures and optimization of storage space.

My study concentrates on defining the different types of objects that can be found in common music notation. This can be seen as the first necessary step in process of object-oriented software development.
One goal of the study was to achieve a simple and coherent model of a system that in itself appears to be extremely complex. To help achieve this goal, I applied systematic methodology beyond conventional object-oriented techniques. In the light of the complexity of music notation as a communication system, the amount of detail that could be handled had to be kept small. Limiting the amount of detail also helps to keep the model and its presentation simple.

IV. RESULT

The results of the study are cantered around a structural analysis model of music notation. The model is presented as a set of UML class diagrams with accompanying text descriptions. The model presents a structure of object classes and their relationships. Once aspect of the analysis involves the classification of musical symbols. Another aspect is the definition of various relationships between the classes. Once central set of relationships is the decomposition of a musical score; this starts from the abstraction of the entire score and proceeds down to its smallest symbolic constituents: note heads, lines, dots, letter, numbers etc.

This is not the first project to apply object oriented techniques to computer based music notation. In fact, several ongoing notation program have been written with object oriented programming languages. Nevertheless, few systematic scientific studies have been published that deal with what music notation is in terms of objects (i.e., classification of music notation symbols). The key contribution of this study is not the new-object model itself. Rather, object-oriented modelling is used here as a means for gaining a new perspective on music notation and on its computer representation. Above all else, the object modelling applied in this study helps to isolate and organize the different types of musical information mentioned above, and to examine their roles and mutual relationships.

What separates this study from previous ones on the same area is that it presents a general purpose model of music notation rather than one aimed at some particular computer program or application. Here, object oriented analysis, further refined with the principles of linear logic, is used as a systematic research method. Moreover, the object-model of

music notation is presented as a result of using this methodology, rather than as a mere case study of some software engineering method. The main aim of this study is to present a description of music notation that can be used as a basis for software design. In addition to this practical aim, the present work can be approached as theoretical study that presents a systematic categorization of the symbols and in Western Music Notation.

The principle value of this study does not lie in the application of object oriented methods to music notation in general. Rather, its main value concerns the way that music notation is approached and how object oriented methods are applied and developed further. Here, music notation is approached primarily as graphical system, which contrasts with some recently developed representations of music notation. Object oriented methodology is extended with linear logic, which provides a strict set of rules to help in forming an object structure for representing music notation.

V. CONCLUSION

Object oriented analysis can serve as a tool not only for software design, but for theoretical examination of a complex system. Although the main intention of this study was to provide a representational basis for developing music notation software, the analysis model can also be regarded in some ways as a theoretical study of the structural relationships of music notation symbols. A central constraint placed on the model was that is be consistent, while respecting the vocabulary and behaviour of the program domain.

The use of a rule set based on linear logic aids in the making of consistent decisions during the analysis stage. Linear logic can, however, be criticized for imposing a stiff and inflexible object structure. Also, linear logic itself has not yet gained wide acceptance in software engineering. For this study, an additional systematic method was needed, so that analytical decisions could be tested against generic and predefined principles, rather than every decision being treated as a special case. For this purpose, linear logic provided an efficient tool. Moreover, it was not presupposed that a programming language based on linear logic would be required for implementing the model.

The graphic notation languages of object oriented methods serve as a convenient and compact tool with which to analyse systems and design software. UML, in particular, is so widely accepted that educated software designers and programmers can be expected to produce analyses and designs even without including explanations of the notation conventions. For this study, only a subset of UML was needed. A description of the subset was included in order to make the text methodologically self-contained.

The key features of the analysis model can be summarized as follows:

1. It presents an object-oriented representation of the problem domain
2. It uses terminology that can be found in the established vocabulary of music notation.
3. It is consistently graphically oriented.
4. It uses a hierarchical class inhabitation structure for categorizing notation symbols
5. It defines a systematic and coherent aggregation structure influenced by linear logic.
6. It suggests that logic information should be modeled as object properties and associations, not as objects
7. It puts little emphasis on explicit representation of purely performance related information.

REFERENCES

[1] Adobe 1986: PostScript Language Reference Manual. Reading, Massachussets; Addison Wesley.

[2] Andreoli, Jean-Marc – Pareschi, Remo 1990: LO and Behold! Concurrent structures processes. ACM SIGPLAN Notices, Proceedings of the European conference on object oriented programming on Object-oriented programming systems, languages and applications. Vol 25, Issue 10.

[3] Asperti, Andrea, Roversi, Luca 2002: Intuitionist Light Affine Logic. ACM Transactions on Computational Logic (TOCL). Vol 3, Issue 1

[4] Bellini, P. Nesi, P. 2001: WEDELMUSIC Format: An XML Music Notation Format for Emerging Applications. First International Conference on WEB Delivering of Music. Florence, Italy: IEEE.

[5] Blostein, Dorothea – Haken, Lippold 1991: Justification of Printed Music Communications of the ACM. Vol 34, Number 3.

[6] Booch, Grady 1994: Object-oriented Analysis and Design with Applications. Second Edition. Santa Clara, California: Benjamin/Cummings.

[7] Byrd, Donald 1994: Music Notation and Intelligence. Computer Music Journal Volume18, Number 1.

[8] Foxley, Eric 1987: Music – A Language for Typesetting Music Scores Software – Practice and Experience 17(8): 485-502.

[9] Grande, Cindy 1997: The Notation Interchange File Format: A Windows Compliant Approach, Beyond MIDI – The Handbook of Musical Codes.

[10] Coad, Peter – Nicola, Jill 1991, Object-Oriented Programming. Englewood Cliffs, New Jersey: Prentice-Hall.

[11] Content Based Image Retrieval System with Regency Based Retrieved Image Library - Jadhav Seema H , Dr.Singh Sunita , Dr.Singh Hari – Paper ID : IJCTT-V19P104

[12] Survey on Big Data: Management and Challenges - K.Indira Gandhi , Sri.C.Sreedhar IJCTT-V20P106

[13] Big Data Analysis: Apache Storm Perspective - Muhammad Hussain Iqbal , Tariq Rahim Soomro - IJCTT-V19P103

[14] Face Recognition Based On Granular Computing Approach and Hybrid Spatial Features - S.Sankara vadivu , K. Aravind Kumar - IJCTT-V20P109

[15] Efficient Handling of Big Data Volume Using Heterogeneous Distributed File Systems - Radhakrishnan R , Karthik S IJCTT-V15P132