WEB-BASED MUSIC NOTATION EDITING

Stuart Cunningham, Nicole Gebert, Rich Picking & Vic Grout
Centre for Applied Internet Research (CAIR), University of Wales
NEWI, Plas Coch Campus, Mold Road, Wrexham, LL11 2AW, North Wales, UK
s.cunningham@newi.ac.uk | nGebert@web.de | r.picking@newi.ac.uk | v.grout@newi.ac.uk

ABSTRACT

MusicXML is a popular and powerful format for representing musical notation and associated data, and is rapidly becoming accepted as the ad-hoc standard for musical information interchange. However, although available under a royalty-free license, MusicXML is mainly supported in commercial products. MusicXML is a very Internet-friendly format, although the applications of this when used in platform specific commercial software are limited.

To fully exploit the capabilities of MusicXML a Flash-based web interface has been developed which not only provides a free, platform-independent MusicXML player and editor, but due to its Internet presence allows music composers to jointly share and co-author musical score. This paper presents the rationale and features of the application, and discusses its potential impact in the user-community.

KEYWORDS

MusicXML, XML, Flash, music, composition.

1. INTRODUCTION

A novel web-based application, provisionally titled *MusFlash*, has been developed to maximise the features of the MusicXML notation format. MusicXML is an increasingly popular, XML-based, file format for musical notation (Cunningham, 2004, Good, 2001, Stewart, 2003). One of the primary goals and intrinsic features of the MusicXML format is its ability to support Internet distribution (Cunningham, 2003, Good & Actor, 2003). However, although MusicXML is now supported in both the major and minor music editing applications we feel that the format has still not achieved its full potential.

Although the applications which support MusicXML are powerful music notation packages, they vary in functionality and cost. For hobbyist or amateur composers this means that quality software with MusicXML support is beyond reach for financial reasons, and lower-priced alternatives do not always provide the functionality required. This leaves a niche for a royalty-free software artefact which supports MusicXML. Furthermore, the existing editing packages do not exploit the fact that XML ports easily to web and Internet applications (Cunningham, 2003, Thuraisingham, 2002). This makes non-Apple and Microsoft computer users disadvantaged when it comes to composing electronically. Additionally, the composer will often be tied to using a particular piece of computing hardware to use their notation software, most often a desktop or laptop computer. This makes composing on the move or away from the office restrictive. A similar concept for a standalone MusicXML application has already been proposed, although not to the same level of detail as in this work (Cunningham, 2004).

We propose from this, the following set of requirements to make MusicXML composition available to the largest market:

A suitable application should:

- be platform independent (ideally web-based)
- impose minimum financial cost
- provide a full range of editing tools

An overview of the major notation packages is presented in Table 1 alongside the MusFlash application.

Table 1. Overview of Popular MusicXML Support (Cost shown in U.S. Dollars)

Software	Platform Independent	Cost (Approx.)	Music Editor
Harmony Assistant 9.2	Windows / Mac only	$70	Yes
Finale 2007	Windows / Mac only	$500	Yes
MusFlash	Yes	Free	Yes
Music RAIN	Yes	Depends on requirements	No
QuickScore Elite Level II v.10	Windows only	$180	Yes
Sibelius 4	Windows / Mac only	$600	Yes

In this table we see that while current software products attempt to meet some of the needs described, there is not a single piece of MusicXML compatible software which meets the full range of requirements.

Currently, the only application which comes close to achieving requirements is the recent Music RAIN application. It is the first web-based product which supports MusicXML. However, MusicRAIN is primarily a *notation playback* tool, and does not allow the user to perform major editing tasks. MusicRAIN is a very powerful tool however, and provides strong evidence to make a case for a Flash-based music *notation editor*.

MusFlash attempts to seize these opportunities and also incorporate a centralised database allowing multiple composers to work on a piece without having to share MusicXML or proprietary files and having to distribute the files themselves.

2. OVERVIEW OF MUSICXML TECHNOLOGIES

The reading of music notation from computer monitors has been an issue in the research domain (Picking, 1997). XML has gained popularity because of its supporting technologies such as the web, web databases, metadata and information retrieval (Thuraisingham, 2002). The MusicXML file format has been documented extensively since its creation (Castan et al, 2001, Cunningham, 2003, Good, 2001, Stewart, 2003). It uses easy to understand XML tags and structures to store a variety of music notation elements, is non-proprietary, and available under a royalty-free licence.

To detail MusicXML extensively is beyond the scope of this paper. However, Cunningham (Cunningham, 2003) provides the simplest of MusicXML examples, the output in a notation reader shown in Figure 1 and the code in Figure 2.

Figure 1. Simple MusicXML Code Example (Reader Output)

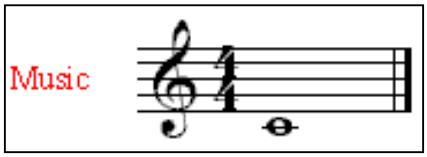


Figure 2. Simple MusicXML Code Example

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC "-//Recordare//DTD MusicXML 0.6 Partwise//EN"
"http://www.musicxml.org/dtds/partwise.dtd">

<score-partwise>
      <part-list>
            <score-part id="P1">
                  <part-name>Music</part-name>
            </score-part>
      </part-list>

      <part id="P1">
            <measure number="1">
                  <attributes>
                        <divisions>1</divisions>
                        <key>
                              <fifths>0</fifths>
                        </key>

                        <time>
                              <beats>4</beats>
                              <beat-type>4</beat-type>
                        </time>

                        <clef>
                              <sign>G</sign>
                              <line>2</line>
                        </clef>
                  </attributes>

      <note>
            <pitch>
                  <step>C</step>
                  <octave>4</octave>
            </pitch>

            <duration>4</duration>
            <type>whole</type>
      </note>

            </measure>
      </part>
</score-partwise>
```

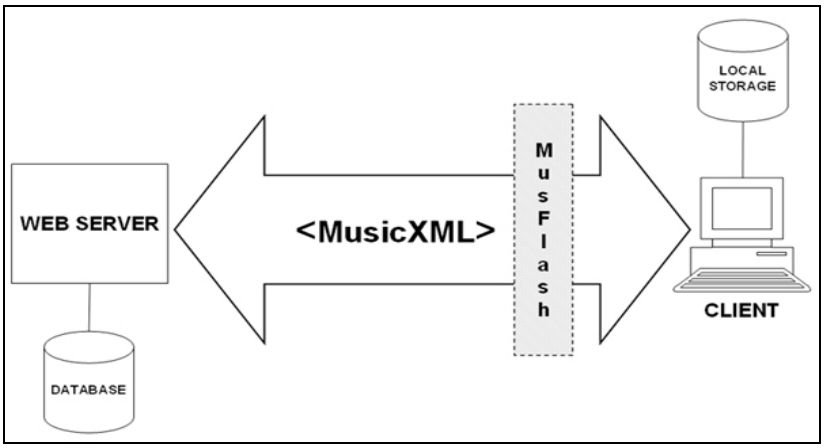
3. THE MUSFLASH APPLICATION

The use of Macromedia Flash to development and distribute our web-based application application was a natural choice. Flash allows any user with a web browser and the Flash browser plug-in to view and use a Flash movie object. The Flash plug-in is free and easy to obtain. The Flash environment allows development of highly graphical, highly functional applications. As XML is an accepted and frequently used tool, predefined functions in Flash Action Script allow opening, reading and creating XML documents.

In fact, an existing piece of MusicXML compatible software has already been developed in Flash. This application uses Flash to produce and rewrite MusicXML, and is known as the MusicRain Project. MusicRain is an online sheet music player which actually uses Flash and a Flash library of integrated .mp3 files to display and play MusicXML. The application allows the user to change the key and the tempo of the displayed music sheet (MediaRain, 2005), but direct interacting with the musical interface is not possible.

Additionally, it is proposed that the MusFlash application integrate with a central web server, which would provide a database of MusicXML compositions by users who are registered on the system. This provides a central location where users can save their compositions and subsequently access and edit them without having to have the file saved locally or even by use the same computer. Essentially this follows a simple client/server paradigm with the inclusion of MusicXML as the data to be transported and the MusFlash application as the interface. An outline of the proposed connectivity of this is shown in Figure 3.

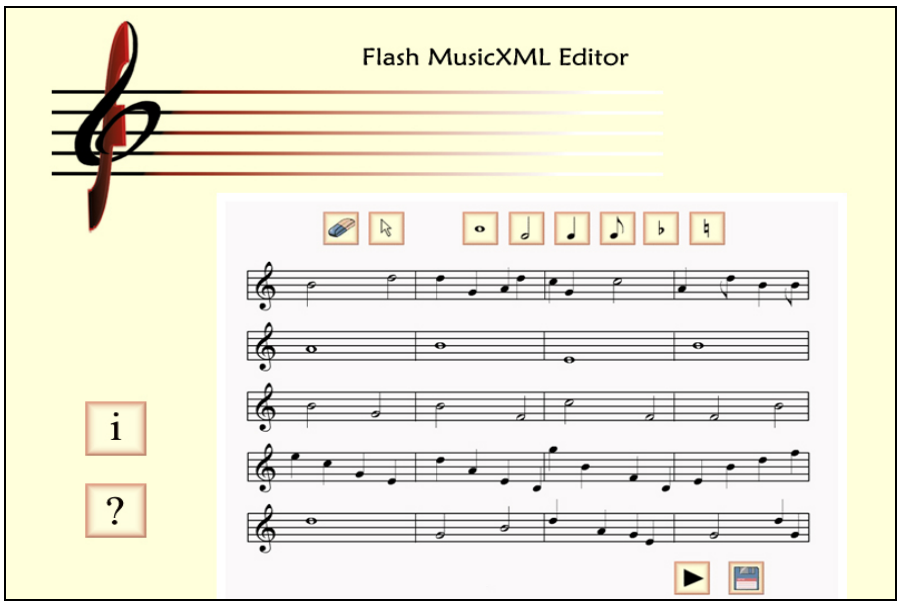
Figure 3. Client Server Model



This presents further advantages such as the ability to support online collaborative composition, as well as making compositions available for others to hear and view. All of this functionality, and more, are supported by the fact that the application is web-based and therefore accessibility and sharing of information become much easier.

Basic functionality is accommodated in the prototype version of the MusFlash application. The user is able to add basic musical notes to a staff and a composition can contain a number of parts. A screenshot of the application interface is provided in Figure 4 to give an impression of the application interface in its current state.

Figure 4. – The MusFlash MusicXML Application



The key backbone framework for the application has been achieved and will be easily expandable to deal with the vast variety of objects supported in the MusicXML format. Currently the application allows a small range of notes to be added and removed as well as providing the sharp, flat, and natural variants of each note type.

4. CONCLUSION

Clearly, MusFlash displays great potential as a solution for musical composition, and provides capabilities that make it far more accessible than some of the current market-leading commercial products. In particular, the architecture and accessibility aspects of MusFlash make it attractive to composers who wish to be uninhibited by the traditional desk or laptop computer.

The main strengths of MusFlash lie in its potential to be used and accessed from almost any Internet enabled device. This means that composers who often experience flashes of inspiration in the middle of the night, or away from traditional notation tools could access the MusFlash resource from a PDA or mobile phone. This is furthered by the ability for composers to share or to co-edit their music if they wish, and also to have the peace of mind that their compositions are safely stored on the MusFlash server.

The basis for the functionality and demonstration of the unique features of MusFlash is present in the current release, although certain options, such as the length, number of parts in the piece, and more unusual musical symbols and ornaments are currently unsupported. These are areas which can be easily expanded with further development time.

REFERENCES

Castan, G., Good, M., and Roland, P., 2001. *Extensible Markup Language (XML) for Music Applications: An Introduction* in *The Virtual Score: Representation, Retrieval, Restoration*, ed. Hewlett, W.B., and Selfridge-Field, E. MIT Press, Cambridge, MA, 95-102.

Good, M., 2001. MusicXML: An Internet-Friendly Format for Sheet Music. Proceedings of XML 2001 International Conference. Orlando, USA.

Good, M., Actor, G., 2003. Using MusicXML for File Interchange. *Proceedings Third International Conference on WEB Delivering of Music*. Leeds, UK, IEEE Press, Los Alamitos, CA, pp. 153.

Cunningham, S., 2003. *Music File Formats & Project XEMO*. MSc Dissertation, University of Paisley, UK.

Cunningham, S., 2004. Suitability of MusicXML as a Format for Computer Music Notation and Interchange, *Proceedings of IADIS International Conference on Applied Computing*. Lisbon, Portugal, pp. III -7.

MediaRain, 2005. MusicRAIN: *Online Interactive Sheet Music Viewer – MusicRAIN 2.0*. MediaRain, Orem, UT, USA. Available from: http://www.mediarain.com/musicrain/

Picking, R., 1997. Reading Music from Screens Vs Paper. *Behaviour & Information* Technology, 16(2), pp. 72-78.

Stewart, D., 2003. XML for Music: A Markup Language That Breaks Down Musical Barriers. *Electronic Musician*, December, pp. 58-64.

Thuraisingham, B., 2002. *XML databases and the semantic web*. CRC Press LLC, Florida, USA.