

# DIAMOUSES

## *An Experimental Platform for Network-based Collaborative Musical Interactions*

Chrisoula Alexandraki, Panayotis Koutlemanis, Petros Gasteratos

*Department of Music Technology and Acoustics, Technological Education Institute of Crete, Rethymnon Crete, Greece*

Demosthenes Akoumianakis, Milolidakis Giannis, Vellis George, Kotsalis Dimitris

*Department of Applied Informatics and Multimedia, Technological Education Institute of Crete, Heralion Crete, Greece*

**Keywords:** Distributed music performance, networked music rehearsal, distributed music improvisation, remote music education.

**Abstract:** DIAMOUSES is an on-going research and development project aiming to establish a platform supporting distributed and collaborative music performance. DIAMOUSES is designed to host a variety of application scenarios, including music rehearsal, music improvisation and music education, thus offering a tailorable enterprise-wide solution for organizations offering network-based music performance services. This paper presents the architectural underpinnings of DIAMOUSES and elaborates on the set-up and the results of a recent pilot experiment in music rehearsal.

## 1 INTRODUCTION

In recent years, music performance (rehearsals, improvisation, musical education, etc) has explored advanced communications-oriented infrastructures and computer-mediated tools in an attempt to facilitate a more social music making. As music performance is a form of expression, it is not surprising that such social music making has striking similarities to conversation in that it involves information sharing, co-presence and multimodal exchanges, as it combines musical signals with verbal and visual cues. Consequently, researchers have embarked on various research strands to investigate tools and possible technological setups for collaborative musical interactions. Specifically, a variety of tools have been proposed such as symbolic representations for music scores (Selfridge-Field, 1997; Legh, 2006; Steyn, 2006; Steyn, 2002; Good, 2001; Good, 2002), music scripting languages (Castan, 2006), toolkits (jfuege) and music notation software (Brandorff et al., 2005), while a range of setups have been explored for group music improvisation (Bryan-Kinns, 2004), instant composition (Jordà, 2001), musical dictations (Tremblay and Champagne, 2007), etc.

On the other hand, substantial work is being devoted on suitable network infrastructures which envisage enabling of remote cooperation among musicians. Barbosa reports that the first efforts in network computer music begin as early as the 1970s with the commercialization of personal computers in the United States (Barbosa, 2003). In subsequent research works the main focus is on exploring structures in which the end-to-end latency can be tolerable for musicians performing the same piece of music. The suggested approaches are concerned either with the use low bandwidth signals, such as MIDI, OSC or other forms of alternative representations for music (Gresham-Lancaster, 1998; Lazzaro, 2001), or the experimentation with bidirectional audio transmission in Local Area Networks and unidirectional audio transmission in WANs (e.g. audience receiving high quality audio orchestral performances through the network) (Sawchuck, 2003).

In this paper our primary target is to report on recent results of a research and development project, namely DIAMOUSES, which seeks to support network based music collaboration in various application scenarios. Our current focus is on describing both the technological set-up and the software tools used to facilitate collaborative music

rehearsals. Rehearsing music in a distributed environment where musicians are physically separated and communicate through the network poses strict requirements on two equally significant issues: a) the reliability of the underlying technological infrastructure and b) the effectiveness of the supporting collaborative environment. The technological infrastructure should provide smooth and low-latency signal transition as well as the ability to transmit audio signals of adequate sound quality. The collaborative environment on the other hand, must provide the means for sharing multimedia objects (e.g. scores, metronomes, etc.) and for enhancing virtual co-presence in collaborative performance.

The rest of the paper is structured as follows. The following section presents the methodology of the project and describes three reference scenarios that have been the starting point of the specification of the DIAMOUSES platform. Then the overall architecture of the system is presented highlighting how DIAMOUSES addresses the challenges set. In the following section, we elaborate on collaborative music rehearsal which constitutes one of the scenarios currently being evaluated. Finally, the paper concludes with a brief discussion of recent results and an outline of on-going and future work.

## 2 METHODOLOGY

The main challenge for DIAMOUSES derives from the intention to establish an integrated platform, allowing for remote collaboration throughout a distributed live music performance environment. Specific technical targets include the setup of the music collaboration environment, the support for innovative forms of musical expression through sensor based interaction and the combination of heterogeneous networks (IP and DVB) in the same platform in order to simultaneously support various routing schemes in signal transition (i.e. broadcasting, multicasting and unicasting).

### 2.1 Requirements Gathering

Gathering user requirements and technical specifications for the DIAMOUSES framework was facilitated by an exploratory survey using questionnaires, site visits, and literature review. Specifically, user requirements were gathered through a questionnaire-based survey targeting two groups of potential users of the system, namely specialists in music (i.e. performers, tutors,

composers, conductors and recording engineers) and the casual users (i.e. the general public) as the potential audience of a distributed music performance. The outcomes of this investigation have been very enlightening both in terms of design and technical specifications.

Regarding the requirements of the music specialists, it was anticipated that they would greatly vary according to the genre of the music performed. In short, the data gathered confirmed that performing classical and contemporary music requires the presence of a musical score whereas musicians of pop/rock and electronic music would generally prefer to perform from memory than having any other means of deciphering the flow of the music. However, musicians of the pop/rock genre had a strong requirement for using an acoustic metronome when remotely performing a piece.

Another important finding of the requirement analysis of the musicians was that in all music genres bilateral visual communication among musicians is considered necessary during performance. In the perspective of the DIAMOUSES platform, this requirement denoted that, in addition to audio, video communication should also be provided to every participant in the distributed performance.

Furthermore, musicians were questioned about their preference in the sound reproduction system used for monitoring the performance. The data gathered indicate a strong preference in using a multi-channel sound reproduction system instead of conventional stereo one. This is consistent with the findings of other studies (Sawchuck, 2003), which conclude that sound reflections from the surrounding area are desirable in the context of a networked performance.

Users belonging to the second group (i.e. the audience) were questioned about their preference in facilities for watching and listening to a distributed music performance, their interest in the content of the provided metadata and the provision of a video-on-demand service relating to the performance. It appeared that users did not have a clear preference on the video display, since they showed equal interest in the alternatives provided, which were television, computer terminal, video projection. However, they showed a clear preference for a surround speaker system (of type 5.1 or 7.1), instead of a conventional stereo one. Regarding the information content of metadata provided, they showed a higher interest in information regarding the music performed (e.g. musicological aspects), rather than information about the participating

performers. Additionally, there was a high interest for video on demand services relating to a distributed performance.

## 2.2 Technical Specifications

The technical specifications for the DIAMOUSES platform were compiled by qualitative analysis of prior research efforts and the up-to-date technologies and software tools on the broader area of multimedia streaming. The general results of this study were that network-based music performances require precise control on minimising the various latencies introduced in the entire process of capture, network transmission and playback of audio signals. Other outcomes of this study include the specification of the network bandwidth demand, the accuracy in stream synchronisation and the tolerance in data errors that are due to network packet loss.

The above requirements depend on the information content of the data exchanged in the network distributed environment. The DIAMOUSES approach deals with three types of data content, namely audio, video and control data. In respect with audio data, CD audio (i.e. raw PCM sampled at 44.1 kHz, 16 bit-resolution, stereo) is the quality threshold for the DIAMOUSES experiments. Regarding video data, DIAMOUSES uses low quality video for the visual communication among the musicians (i.e. MPEG-4 or MJPEG format), in order to compensate for bandwidth demand, and high quality MPEG-2 for the communication with the audience in the digital TV infrastructure. The term 'control data' is used to refer to the various symbolic sound and music representations, which may be used as a low bandwidth alternative for uncompressed audio. The project is primarily investigating control data derived from MIDI instruments and experimental mechanical sensors, which are being used for providing gestural control to electronic instruments (Wright, 2001).

## 2.3 Reference Scenarios

To streamline the design of DIAMOUSES, scenarios were used to provide early reflections and envisionment. Scenario elaboration focused on a range of criteria with the most prominent being the operational intent (i.e. live performance, rehearsal, master class or remote studio recording session); the underlying roles undertaken by the participating users (i.e. performers, conductors, composers, tutors, engineers) and the type of music being performed. Additional parameters, which affect the technical

requirements of an application scenario, include the content of the data (audio, video, control) exchanged among participants and the underlying network infrastructure (i.e. wireless or wired, local or wide area networks).

Although DIAMOUSES is implemented as a generic platform to enable various application scenarios with diverse requirements, the evaluation of the system under development will be based on three well defined scenarios. The selection of these particular scenarios, which are described in the remaining part of this section, was based on four criteria. The selected scenarios should (a) combine as many of the interaction parameters as possible; (b) comply with the results of the user requirement analysis, (c) they should be technically feasible to achieve with the existing network infrastructures; and finally (d) they should be scalable in terms of the number of users participating in the scenario as well as in terms of the geographical spread of the connected participants. Scaling the geographical spread of an application scenario is mainly related to the efficiency of the underlying network infrastructure. Since the entire implementation of DIAMOUSES is independent of the lower layers of the TCP/IP network model, the scenarios we define here can be scaled to a broader geographical spread when appropriate network infrastructures become available.

### 2.3.1 Distributed Music Rehearsal

The first scenario to be evaluated concerns a distributed music rehearsal in the range of a Local Area Network (LAN). In this scenario, musicians communicate via a streaming server with high quality audio streams. The server is responsible for delivering the audio signals from each musician located at a certain network node to the other musicians that have been registered as the recipients of the audio stream transmitted from the first node. This mechanism for information exchange is based on a so called 'relay network'. In addition to audio, video streams are exchanged among the participants. The distributed music rehearsal scenario is further elaborated in the next sections of the paper.

### 2.3.2 Distributed Music Performance

The second scenario concerns a distributed music performance experiment based on control data and video data. Participating musicians are connected through a Wireless LAN. MIDI data and OSC data are exchanged in the local network through the

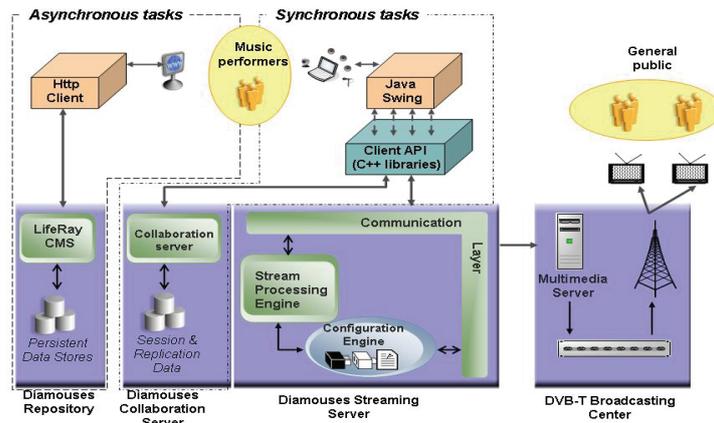


Figure 1: Diamouses overall architecture.

server. The audio synthesis (i.e. the conversion of control data to audio) takes place at the receiving end through appropriate hardware and software equipment (i.e., tone generators). Visual communication among participants is provided through low quality video. Furthermore, as depicted in figure 1, this scenario utilises a pilot digital TV platform which is exploited in the DIAMOUSES framework. The subscribers of this DVB-T network may receive the multiplexed video and audio streams on their TV receivers through a live broadcast at the time of the performance, or through a video-on-demand service after the end of the performance.

### 2.3.3 Remote Music Education

The third experiment aims to examine a remote music education scenario based on the DIAMOUSES platform. In this scenario a music tutor is connected through the Internet with distributed members of a master class and they communicate using audio and video streams. The research focus in this scenario is placed on exploring the interfaces that are necessary for conducting a remote music lesson (theory), exercising with replicated visual representations of music scores and finding the appropriate quality of the audio stream that can be transferred on the Internet with tolerable delay and data loss.

## 3 DIAMOUSES ARCHITECTURE

Figure 1 depicts the overall architecture of the DIAMOUSES platform. As shown on Figure 1,

music performers' collaboration may be either synchronous, during a collaborative performance, or asynchronous, in order to schedule their performance events or share various sources of information. Synchronous tasks are carried out through the communication with a Streaming Server, whereas asynchronous tasks are provided through a portal, which is supported by a Content Management System (CMS). As shown on the right-hand side of the figure, the audience of the synchronous performance is served through a digital TV network.

The DIAMOUSES Streaming Server consists of three main modules: the communication layer, the configuration engine and the stream processing engine. The communication layer is responsible for receiving multimedia streams from all the connected participants and transmitting them back either as a combined stream (in the case of multicasting) or as separate streams (in the case of a relay network).

The configuration engine takes care of the various configurations that need to be done on the server (i.e. broadcasting, multicasting, unicasting). Among other things, it defines the network addresses and the network ports for the incoming and outgoing multimedia streams (i.e. the transmitters and the receiver of the information).

The stream processing engine is responsible for all the processing that needs to be done after a stream is received at the server and before it is sent back to the participants that have been registered as receivers of this stream. Depending on the scenario, stream processing may include stream synchronisation, stream mixing or multiplexing, error correction and data storage of the incoming streams in multimedia files. These files are necessary for allowing DIAMOUSES to offer video-

on-demand services through the DVB platform, after the end of the live broadcast.

The component denoted as Client API in Figure 1 is the low level application software running on the performer side. This software is responsible for capturing audio video and control streams, transmitting these streams to the network, receiving the streams representing the contribution of the other musicians and finally reproducing the streams on the client hardware (i.e. soundcard, speakers, and computer screen). This component is implemented as a low level module aiming at minimising the latencies that are introduced in the various phases of the above process. The Client API communicates with a higher level component developed in Java Swing, which provides the performer with the GUI shown on Figure 2.

The DVB-T broadcasting centre constitutes the pilot digital TV platform exploited by the project. It has a one-way communication with the DIAMOUSES Streaming Server. The purpose of this communication is to send the multimedia streams of the combined performance of the distributed musicians to the subscribers of the digital TV network during a live broadcast. In addition, the streaming server communicates with the DVB-T centre in order to send the multimedia files stored in the streaming server, so that they are available for offering video-on-demand services. All the processing of the audio and video streams (i.e. transcoding, multiplexing, etc.) that is necessary prior to broadcasting occurs at the DVB-T broadcasting centre.

## 4 COLLABORATIVE MUSIC REHEARSAL

DIAMOUSES groups engaging in music rehearsals can be seen as 'local' communities of practice. Although such communities share common ground with conventional web communities of interest, they are also characterized by distinct features. Specifically, they are task-specific and knowledge-intensive in the sense that members need to possess common ground, share data of various types, and contribute towards the creation of new knowledge (codified musical experiences). Moreover, the community resources vary from traditional document-based materials to multimedia objects (i.e. video, sonic objects, etc). In this effort both personal and group learning prevails as the prime social objective. Moreover, accomplishing the common

goals is strongly dependent upon situational aspects and the wider community context.

Music has traditionally served as a natural mechanism for community formation. Indeed music choice frequently serves as a means for community formation and establishing identity (Frith and Goodwin, 1990). Moreover, in recent years a number of 'music communities' have been formed to facilitate promotion and distribution of music. In both these cases, the community serves as a common place for communication, exchange of opinion and sharing of (pre-recorded sonic) artefacts.

In DIAMOUSES the notion of 'music community' holds a slightly different connotation. DIAMOUSES communities are virtual social spaces for conducting collaborative work. In this context, communication and sharing may be important, but they are not the only tasks to be supported. DIAMOUSES communities can be considered as 'squads' formed around a digital medium to achieve a common goal (i.e. rehearsal, live performance, learning, etc). In doing so, each member of the squad undertakes a distinct role which is characterized by social as well as task-specific parameters. Such roles include the conductor, the musician, the observer, all having distinct responsibilities and access rights. Goal achievement assumes willingness to participate, while it is primarily dependent upon the collaborative contributions of all partners. Such contributions in turn utilize resources which can be classified as generic or shared (i.e. music scores) or neighbourhood-specific (i.e. musical instruments). To attain the goal, the group undergoes distinct stages such as formation (i.e. registering as members of the squad), establishing common ground by sharing information, exchanging opinion on certain aspects of the task (i.e. range of musical instruments, tempo), setting norms (i.e. what is to be done, when and how) and finally performing towards accomplishing the common objectives. These stages are typically translated into distinct tasks of two primary types, namely personal/synchronous tasks carried out by members during music rehearsals and community-oriented/asynchronous tasks carried out prior to or after music rehearsals. As these tasks are distinct, different tools have been designed to support them.

### 4.1 The CMS for Asynchronous Tasks

Community-oriented / asynchronous tasks are hosted by a suitably customized version of the LifeRay Content Management System. This is an open-

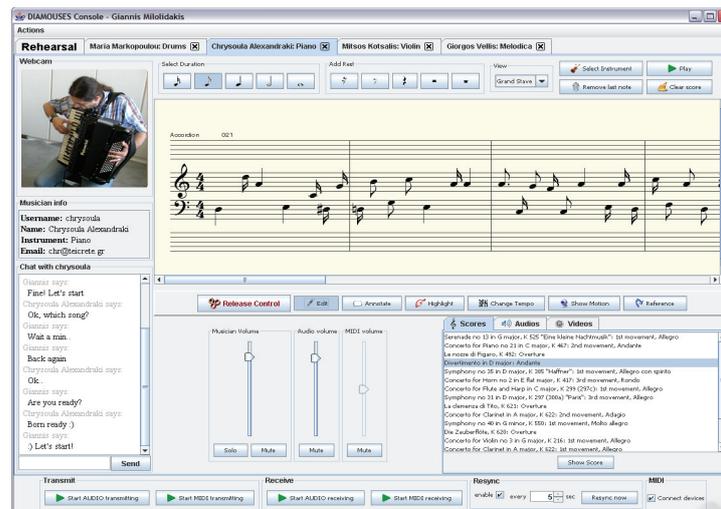


Figure 2: The user interface of the rehearsal application.

source Java based CMS which makes use of portlets for content management. Access to the contents is role-based. Thus, visitors can create an account using the LifeRay registration system, manipulate this account and obtain access to the CMS content. Conductors and musicians represent roles with additional access rights. Specifically, they are entitled to download application suites which allow them to take part in music performances (i.e. rehearsals, improvisation, music education, etc). To facilitate access to these tools, an electronic registration system has been developed using Ajax to allow these advanced roles to register and obtain access to the additional resources. A dedicated portal has also been developed as a container for rehearsals. It is open for registered participants and acts as a reference point, hosting shared objects and sonic artefacts generated and deposited by the group.

## 4.2 Synchronous Rehearsal Application

To take part in a rehearsal, all partners use a dedicated application depicted in Figure 2. It should be noted that the version of the user interface shown on Figure 2 represents a superset of what is actually needed and used during music rehearsals. This is because the application is designed having in mind the requirements of the other reference scenarios. This application is separate from the CMS to alleviate latency problems. During a rehearsal, each participant has visual access to a remote site through a camera and can control various aspects of the remote user's performance such as the volume of the signal received from a remote site, by using the sliders at the bottom-centre. It turned out that this

was the dominant tasks during the rehearsal. However, participants can carry out additional tasks. Specifically, they may choose to view the musical score which occupies the middle part of the screen and is uploaded from the corresponding CMS room. This is a 'shared' object which can be manipulated either in private or through synchronous sessions. In case of synchronous sessions (foreseen primarily for the remote music education scenario) the object is replicated across the connected clients and remains synchronized at all times. This is facilitated by a floor manager undertaking runtime permission assignments. At any time, participants can exchange messages (i.e. chat) and explore background materials (i.e. musical scores, audio files, video, etc) deposited as shared resources in the corresponding CMS room. The graphical user interface (GUI) is implemented using Java Swing and dedicated Java music APIs. The communication of the Client API and the GUI is implemented in Java Native Interface (JNI).

## 5 THE PILOT SET-UP

The pilot music rehearsal scenario was conducted in the Department of Music Technology and Acoustics (MTA) of the T.E.I. of Crete in October 2007. For the purposes of the rehearsal scenario, three professional musicians were invited. The scenario involved rehearsal with both acoustic instruments and MIDI instruments. The musical piece and the music instruments used in the rehearsal were a choice of the musicians. They chose to perform a Greek song ("Mia thalassa mikri", Dionysis

Savvopoulos), with accordion, guitar and saxophone. The same piece was rehearsed with acoustic accordion guitar and saxophone as well as with a MIDI accordion, a MIDI guitar and a MIDI saxophone.

### 5.1 Musician Apparatus

Each of the three musicians was connected to the network through a portable PC running Linux distribution Centos 5 (see Figure 3). In addition to the portable PC, each musician was equipped with an external USB soundcard (for providing better audio quality than the quality provided by the laptop built-in soundcard), a microphone, a MIDI tone generator, a pair of speakers and a network camera. A sound console was used for allowing simultaneous playback of the output of the soundcard (i.e. the audio streams arriving from the network), and the output of the MIDI tone generator (i.e. the MIDI streams arriving from the network, after converted to sound). This was occasionally necessary for the musicians in order to talk to each other while tuning their MIDI instruments. The experiment was broadcasted through wall projection of the live video feed from the three musicians (see Figure 4) to a number of group members watching the experiment. Figure 5 presents one of the musicians (namely the accordionist) during the audio rehearsal experiment. Video communication among musicians was provided through the GUI of the client software (Figure 2).

### 5.2 Network Topology

The three musicians were distributed in three different buildings in the campus network of the MTA Department and they were connected through the DIAMOUSES Streaming Server, which was located at the same building as one of the musicians. The provided network connection was a 100Mbps Local Area Network connection. For the streaming server a customised version of Darwin Streaming Server was used, running on Red Hat Enterprise Linux 5.



Figure 3: Musician apparatus.



Figure 4: Video projection of the musicians.



Figure 5: One of the partners during rehearsal.

### 5.3 Data Transmission

As mentioned before, two experiments were conducted: the “audio rehearsal” and the “MIDI rehearsal”. In the audio rehearsal, each musician was sending to the server the audio stream captured through their microphone. The server was relaying back the audio streams captured from the microphones of the other two performers. Mixing was performed on the client side. In addition to the audio streams, each musician was receiving the live video stream captured from the network cameras of the other two performers. It was found that video quality had to be reduced in order to avoid distortion in the audio stream, which was perceived as clicks (data loss).

In the MIDI rehearsal, each musician was sending the MIDI stream produced at their MIDI instrument, which was connected to the USB soundcard. The client software was sending this MIDI stream to the server. The MIDI streams of the other two performers which were relayed from the server back to the musician, were fed to the MIDI

tone generator through the USB soundcard. Video streams from the other participants were received at the site of each performer directly from the network cameras. In the MIDI experiment, no data loss was observed when raising the video quality. Table 1 following table denotes the data formats used for the multimedia streams, their quality characteristics and the resulting data rates.

Table 1: Data transmission rates.

Data	Audio	Video	Control
Format	Raw PCM	MJPEG	MIDI
Quality	48kHz 16-bit stereo	320x240 14fps	N/A
Data rate	1.536 Mbps	630kbps	31.25ks

## 6 DISCUSSION & CONCLUSIONS

The analysis of data gathered during the experiment provided valuable insights to the task of conducting network-based music rehearsals. Specifically, the overall latency in the transmission of audio and MIDI signals was not noticeable from the musicians and it was estimated in the order of 10msec. With regards to data loss, it turned out that there were no significant errors when the bitrate of the video streams was kept low. Raising the quality of the video streams resulted in distortion in the audio signal playback, which we presume was due to network packet loss. Another technical issue that was causing problem in the communication among musicians was the fact that when one of the musicians was executing a program change event at a certain MIDI channel, the event was propagated through the network to the equipment of the other participants. This undesirable effect implies that MIDI data should be filtered before sent to the server or before arriving to the target receivers.

Overall, musicians were excited about the experience they had, and they said that they would be glad if they had the opportunity to use this platform for rehearsing with their band from home. The final evaluation of the system will be based on both QoS measures and on user feedback. We expect to measure latency and data loss in various configurations. In addition, as there is evidence that both tempo and acoustic properties of the instruments (i.e. timbre) greatly affect the success of the experiment (Sawchuck 2003), we plan to experiment with different types of music and instruments.

## REFERENCES

- Barbosa, A., 2003. Displaced Soundscapes: A Survey of Network Systems for Music and Sonic Art Creation, *Leonardo Music Journal - Volume 13*, MIT Press, 53-59.
- Brandorff, D., Lindholm, M., Christensen, HB., 2005. A Tutorial on Design Patterns for Music Notation Software, *Computer Music Journal*, 29:3, 42-54, MIT.
- Bryan-Kinns, N., 2004. Daisystone: The Design and Impact of a Novel Environment for Remote Group Music Improvisation, in *Proceedings of ACM DIS'2004 Conference*, August 1-4, 2004, Cambridge, MA, USA.
- Castan, G., 2006. MusiXML, <http://www.music-notation.info/en/musixml/MusiXML.html>.
- Good M., 2001. MusicXML Definition. <http://www.recordare.com/xml.html> [2001].
- Good M., 2002. MusicXML in practice: Issues in translation and analysis. *Proceedings of the 1st International Conference MAX2002 Musical Application Using XML*, Milan, Italy, September Computer Science Department, State University of Milan: Milan, 47-54.
- Gresham-Lancaster, S., 1998. The Aesthetics and History of the Hub: The Effects of Changing Technology on Network Computer Music, *Leonardo Music, Journal 8* (1998), 39-44.
- Jordà, S., 2001. New Musical Interfaces and New Musicmaking Paradigms. *New Instruments for Musical Expression Workshop*, (Seattle).
- Lazzaro, J., Wawrzynek, J. 2001. A Case for Network Musical Performance, in *Proceedings of ACM NOSSDAV '01*, NY, June 2001, 157-66
- Lugh, M., Montgomery, L., 2006. 4ML a Music & Lyrics Markup Language. <http://www.4ml.org>.
- Selfridge-Field, E., 1997. *Beyond MIDI: The Handbook of Musical Codes*. MIT Press: Cambridge, MA.
- Steyn, J., 2006. Music markup language. <http://www.musicmarkup.info> [June 2006].
- Steyn J., 2002. Framework for a music markup language. In *1st International Conference on MAX2002 Musical Application Using XML*, State University of Milan, 22-29, Milan, Italy.
- Tremblay, G., Champagne, F., 2007. Marking musical dictations using the edit distance algorithm, *Software Practice & Experience*, 37:207-230
- Sawchuk, A.A., Chew, E., Zimmermann, R., Papadopoulos C., Kyriakakis, C., 2003. From Remote Media Immersion to Distributed Immersive Performance, in *Proceedings of the ACM SIGMM 2003. Workshop on Experiential Telepresence*.
- Wright, M., Freed, A., Lee, A., Madden, T., Momeni, A., 2001. Managing Complexity with Explicit Mapping of Gestures to Sound Control with OSC. *Proceedings of the International Computer Music Conference*, Habana, Cuba.