Public Sound Objects: A Shared Musical Space on the Web

Álvaro Barbosa, Martin Kaltenbrunner
Music Technology Group - Pompeu Fabra University
Passeig de Circumval·lació 8 - 08003 Barcelona, España
E-mails: {abarbosa; mkalten}@iua.upf.es

Abstract

In this paper we describe "The Public Sound Objects" project and its context. This project, which is currently under development, approaches Internet collaborative music performance, going beyond most common paradigms where the network is mainly used as a channel to provide a connection between performative spaces. The system will provide an on-line public performance space where people can be found participating in an ongoing collaborative sonic event. The users connected to this installation control a server side synthesis engine through a web-based interface. The resulting "Sound Objects" form a sonic piece that is then streamed back to each user. The user takes the role of a performer and his contribution has a direct and unique influence on the overall resulting soundscape. This ongoing event is also played back at the installation for a live audience, with added contextual elements such as sound spacialization and metaphorical visual representations of current participants.

Keywords: Sonic Arts; Internet Music; Shared Virtual Environments; Web based Collaboration; Sound Synthesis; Human Computer Interaction; Internet Collaborative Music Performance.

1. Introduction

Computer-Supported Cooperative Work (CSCW) is one of the major research fields in modern information society, and recent technological advances, specially in Internet computing, have allowed computer science researchers and developers to create different types of collaborative tools, such as white boards, shared editors, video conference systems or even e-mail based systems that are already part of our daily life.

On the other hand during the last decades we heave seen artists taking cutting edge technology and using it to maximize the aesthetics and conceptual value of their work, not only by enhancing the way they traditionally create, but also by using technology as a media itself to express meaningful artistic work.

The idea of using computer networks as an element in collective artistic creation and performance (or when both come together in improvisation) was no exception.

Collaboration paradigms have great relevance in music, since traditionally music performance is a result of joint synchronous events where musicians with their individual performance contribute in real time to a final piece.

Early experiments with musical computer networks at a local area scale date back to the late 1970's in California with performances by the League of Automatic Music Composers [1].

However with the massive world wide growth of the Internet community, characterized by users strongly moved by music in many different ways, more appealing possibilities for music composers and performers came up in the 1990's.

1.1 Collaboration over the internet

So far in music or sonic arts collaboration experiments over the Internet the biggest breakthrough has been the capability to provide remote communication between worldwide displaced musicians and composers. This type of connectivity tremendously enhances the traditional collaboration paradigm for music production.

Early experimental systems based on this idea go back to the early 1990's with the Craig R. Latta's NetJam [2] from Berkley University. This system allowed a community of users to collaborate producing music in an asynchronous way by automatically exchanging MIDI files trough e-mail.

Other experimental systems focused more on the idea of having synchronous performances as close as possible to a real-time situation, like the 1998 *TransMIDI* [3] system, implemented using the *Transis* multicast group communication layer for CSCW applications [4], or Phil Burk's *TransJam* [5] that also allows this kind of interaction but going beyond the MIDI format and allowing low fidelity digital audio.

Recently commercial systems based on client-server architectures allow the collaboration on music pieces using MIDI and digital audio formats. Systems such as the *ResRocket Surfer* [6] or the *Tonos* system [7] have been highly successful receiving reasonable support from music industry manufacturers.

All these systems provide effective enhancements in the process of music production, however, they are mostly oriented towards a traditional studio production environment, leaving little space for more experimental forms of Sonic Arts, and constraining the potential of what the Internet can offer as a media for artistic expression.

Very few examples can be found where the Internet's possibilities, more than just allowing remote connections between two traditional events, were embraced by the artists as elements that actually contribute to their piece.

1.2 More than just Tele-Presence

Remote performance in a live public event incorporating low-cost public domain technology is one of the most appealing possibilities provided by the Internet.

Many public events with remote presence of musicians over an Internet connection have been performed in the last few years.

Different styles of music, instruments and technical setups have been tried like the *Telemusic* and the *Piano Master Classes* by John Young and Randall Packer [8] [9], the New York University's Cassandra Project [10] or Robert Rowe's demonstration of *Real-Time Internet Multi-Channel Audio* during the 107th AES Convention [11].

Situations of remote performance raise interesting questions about performance techniques and what should be the actual remote performer's visual and sonic representation on site.

In any case serious considerations should be made when integrating the inevitable network delay as an element into the resulting sonic soundscape, like we can find in the work of Chris Shafe and Greg Niemeyer in the *Ping* sonic installation project developed at Stanford University [12].

A more complex scenario than a unilateral remote performance is a performative collaboration between two or more simultaneous events. In October 2001, during the *Networkshop* festival in Dresden, Germany, several collaborative on-line concerts based on the FMOL Virtual Music Instrument [13] took place between there and Barcelona.

The concerts consisted of improvised duets, using a peer-to-peer version of the new FMOL system [14], which supported real-time network jamming. Attained delays were in the range of 100 ms using a conventional 56 kb modem connection, providing a very good feeling of playability. This condition of immunity to network delays in FMOL music is related to the nature of the resulting soundscape.

The sound sequencing technique used in this system, based on low frequency oscillators (LFO) excitation of sound generators, creates rhythmical and melodic progressions that to some extent support flexible reaction times and short lacks of synchronicity from the performing partners.

A different approach in collaborative Internet performances is to allow free access for the Internet community of *CyberNauts* as performers in public events for live audiences.

For a regular Internet user, having the possibility to perform over the Internet with one of Tod Manchover's *Hyper-Instruments* during a Brain Opera session in Vienna's *Haus der Musik* is a key factor for the highly successful results of these artistic proposals. Collaborating with others in Jorda's FMOL [15] over the Internet in a piece that could be selected for the music score of La Fura dels Baus's *Faust 3.0* opera premiered in Barcelona's Teatro del Liceu was equally appealing.

Another approach for internet based collaboration in a mixed environment between a museum and an on-line public installations is the Worldbeat/GlobeMusic project [16] currently hosted by the Ars Electronica Center in Linz. This project as been quite successful as a novel computer based musical instruments local installation, however realizing that playing music becomes more interesting when several people join to play together, it's creators are developing ways for offline/online interaction with other musicians (WorldBeat: NetMusic).

In this collaboration paradigm the "well-sounding music" and "being-in-control" experience of the users will be preserved, since it was proven to be the key factors for the big public success of the WorldBeat system according to systematic evaluation with users.

The concept of community-oriented music itself is also an interesting open research topic. It is yet to be clarified until which extent the average internet user is prepared to participate in a creative process, contributing meaningfully to an artistic event and what kind of constraints should be considered when designing such systems.

A very fruitful discussion related with this topic was held in December 2001 in the interactivity discussion panel during the MOSART Workshop in Barcelona [17].

1.3 Shared Musical Spaces

More recently other proposals of Sonic Art in the context of community-oriented music focused on collaboration in expressive Internet sound events exclusively for Internet audiences.

These proposals are shared musical spaces where people can be found performing in collective music pieces, given that everyone should be able to choose either to participate as a performer or simply as a member of the audience.

Pioneer Internet systems that convey the essence of a virtual community space are based on the original MUD ("Multiple-User Domain/Dungeon") software developed in the early 1990's by Pavel Curtis, Xerox Corporation [18]. In a MUD or in its successors like the MOO (MUD, Object-Oriented) or the IRC (Internet Relay Chat), the participants (usually called players) tend to develop a specific language for communication and collaboration amongst themselves, that evolves to some sort of virtual social behaviour, that only makes sense in these environments [19].

William Duckworth's 1997 Internet based Cathedral [20], is one of the first interactive music works created specifically for the World Wide Web. Other relevant examples of work developed in this context are the *WebDrum*, based on Phil Burke's *Javasyn* [21], or Atau Tanaka's *MP3Q* piece on the web [22].

2. The Public Sound Objects

The *Public Sound Object* project is being developed in the Music Technology Group of the Pompeu Fabra University.

The project shares common ground with many of the previously mentioned proposals for Internet based Collaborative Virtual Environments focused on sonic arts and music creation. However, in our approach we explore the concept of a shared musical space in the sense of community-driven music creation, as in an art installation that brings together both a physical space and virtual presence in the Internet, allowing synchronous interaction amongst web users

The overall system architecture is designed along the following key aspects:

- It is a public event with characteristics that should be appealing both to a "real world" live audience and for a virtual audience of occasional *Cybernauts* visiting our server.
- The tele-performers' contribution to the final musical piece should be adequately constrained in a way that the overall aesthetical coherence of the piece can be guaranteed.
- The system should be scalable and modular enough to allow future extension and further experiments with different setups.

Besides the system implementation, in this paper we also discuss a proof of concept interface and on-site installation prototype that we are designing and implementing in parallel.

2.1. An ongoing public event

One of the questions that stand out when designing a system with these characteristics is whether it is reasonable to consider a music piece as an event limited in time?

In fact until now most of the artistic proposals for public events designed for community performance - even the most acknowledged art pieces like the *Vectorial Elevation* by Rarrael Lozano-Hemmer [23], awarded with the Prix Ars Electronica for interactive Art in 2000 - have been developed towards an event that takes place at a specific date during a certain period of time, when the presence of a physical and/or virtual audience in a theatre-like experience is guaranteed.

We argue that it is the Internet's essence to provide permanent connectivity, thus it makes sense that a public Internet event should go on permanently, and that the audience and performers are free to join and leave at any time they want.

Therefore this event is permanent and public, since it is continuously displayed to the public both via the Internet and on the installation site discussed later in this paper. It also provides the permanent possibility for the public to choose either the performer's or the spectator's role.

2.2 Sound Objects

In this project the raw material provided to the users for their contribution to the performance are Sound Objects. The definition of a Sound Object as a relevant element of the music creation process goes back to the early 1960's [24]. According to Schaeffer's, a Sound Object is defined as "any sound phenomenon or event perceived as a coherent whole (…) regardless of its source or meaning" [25].

Although there are advantages in using logical formats like MIDI from the communications point of view in distributed sound systems, defining the universe of sound events by subsets of Sound Objects is a very promising alternative for content-processing and transmission of audio [26]. Entering a session in our system a user can choose from a set of Sound Object provided as sound samples.

From a psychoacoustic and perceptual point of view, Schaefer's definition is extremely useful, since it provides a very powerful paradigm to sculpt the symbolic value conveyed in a sonic piece. The symbolic value of the Sound Object is a key element for the construction of sonic soundscapes.

Adding metaphorical value to a Sound Objects enables the user to identify it within the piece. On the other hand the symbolic value of the Sound Object might also change depending on the context where it is presented.

In many applications such as Auditory Users Interfaces, Sound Objects must be simple and straight forward, so that there is no ambiguous understanding of what they intend to represent. However in an artistic context the scope for the user's personal interpretation is wider, therefore such Sound Objects can have a much deeper symbolic value and represent more complex metaphors.

Often there is no symbolic value in a sound, but once there is a variation in one of its parameters it might then convey a symbolic value. A typical example is the use of white noise to synthesize wind sound. If we listen to continuous white noise it might not represent a very strong metaphor, although we could relate it with some meaning depending on its context. It can for instance be perceived as an offline transmission device.

However, if we apply a band pass filter to this sound, varying its central frequency, even without any special context we can perceive the result as the very familiar natural sound of wind blowing.

In our system a server-side real-time sound synthesis engine provides the interface to transform various parameters of a Sound Object, which enables the user to a certain extent to add symbolic meaning to his performance.

3. System architecture

As shown in the illustration below, the *Public Sound Object* system is based on classic client-server architecture. The actual sound synthesis computation is handled by the server and the interaction interface is implemented on the client side. One of the main characteristics of this implementation scheme is its modularity.

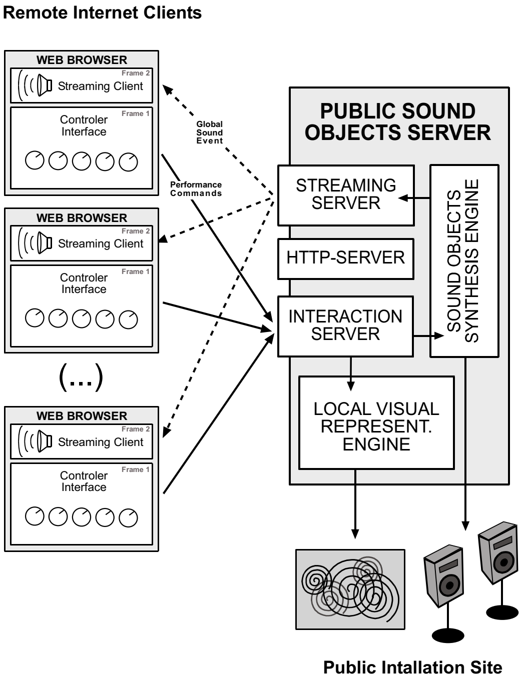

Fig. 1 -The PSO Architecture

The main application at the server side is the synthesis engine, which is designed in a rather general way in order to allow its versatile use for different applications. This engine is configured and controlled by two interfaces: a configuration interface, which initialises the general sound installation set-up by reading from a configuration file and a real-time control interface which allows an external application to control the various parameters of the synthesis process during execution time.

The core technology of this synthesis engine is based on CLAM (C++ Library for Audio and Music), a set of audio synthesis C++ Classes, designed at the Music Technology Group in Barcelona. It allows flexible sound transformations, by providing a versatile interface for the modification of a large number of audio parameters.

Conceptually the engine is a re-implementation of Serji Jorda's FMOL synthesiser engine.

The second server-side module is the interaction-server, which basically manages the sessions of the various connected users. It processes the received interface parameters and controls the according sound object synthesis processes within the synthesis engine. Additional components are a streaming audio server, which broadcasts the final audio stream back to the users and a standard HTTP server for delivering the web based interface.

Finally there is also a server side graphics component for the creation of the on-site visual representation of all current user sessions. The public sound server is part of a physical sound installation located in an appropriate museum space.

The resulting piece is played back by a local speaker system, and a visual representation of the installation is projected. The installation site provides as well some local client machines to allow local visitors the spontaneous participation in the piece.

On the user side the main application is a Java applet, embedded into the web interface. This applet upon loading connects to the interaction server, registers and initialises a user session.

It provides the complete graphical user interface for the interactive control of the synthesis process. An additional component is the streaming audio client for the playback of the resulting musical piece.

3.1. The Synthesis engine

The synthesis engine incorporated into our system is a re-implementation of Sergi Jorda's original FMOL synthesizer. This project, currently carried out at the MTG in Barcelona, aims not only to re-implement but extend Jorda's synthesis concept using the CLAM C++ framework.

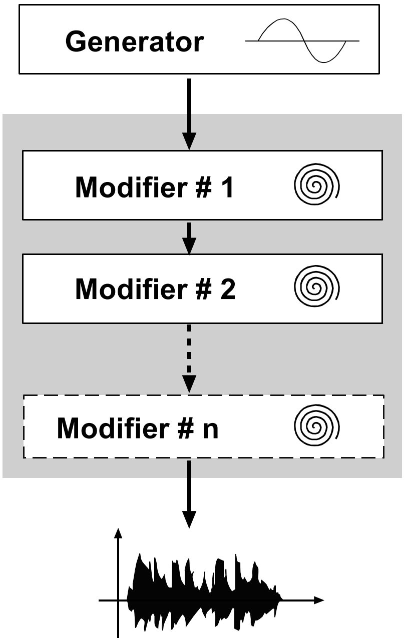

Fig. 2 -The Synthesis Engine

Basically this synthesizer provides a sound generator which ranges from basic oscillators and modulations up to the simple use of digital sound samples.

This generator's waveform is then altered by a chain of at least three modifiers. These modifiers implement a large toolbox of digital filters such as IIR or comb filters and effects such as pitch shifting etc.

These chains of modifiers provide a suitable amount of alterable parameters for our idea of mapping the interaction with an abstract visual model to the actual Sound Object. Finally the FMOL synthesizer can handle a large number of such tracks - those generator/modifier models which each actually represents an individual Sound Object. These independent sound tracks are then mixed together to the final musical piece.

3.2. The user interface

The user interface allows the interaction with the server-side synthesis engine and focuses on the manipulation of the actual sound synthesis parameters.

Due to the modular nature of the system the component that is likely to vary in different setups is the graphical user interface (GUI).

In our system each GUI implementation, called Skin, will be developed along the following requirements:

- It should enable the user to contribute to the ongoing musical performance by transforming the characteristics of a visual Sound Object representation, sending normalized parameters to the synthesis engine over the network.

- The interface application should be able to manipulate the parameters for each of the modifiers in the synthesis engine according to the specific installation site setup.

- The GUI itself should be a behaviour–driven metaphorical interface, avoiding a flat mapping of parameters in a classical way, such as faders or knobs. The Sound Object representation has a default automatic periodical behaviour that can be conducted by the user.

- The auditory feedback conveys the performance of all currently connected users. Optionally there can be added some local auditory feedback which is not part of the actual piece (the same way one would use a metronome).

As a proof of concept application we are currently developing a prototype based on a bouncing ball skin.

The interface, shown below, is a metaphor for a ball that infinitely bounces on the walls of an empty room.

When the ball hits one of the walls a network message is sent to the central server where the corresponding Sound Object is triggered, and played trough a specific source speaker and simultaneously streamed back to the user in a stereo mix of all the sounds being triggered at the moment.

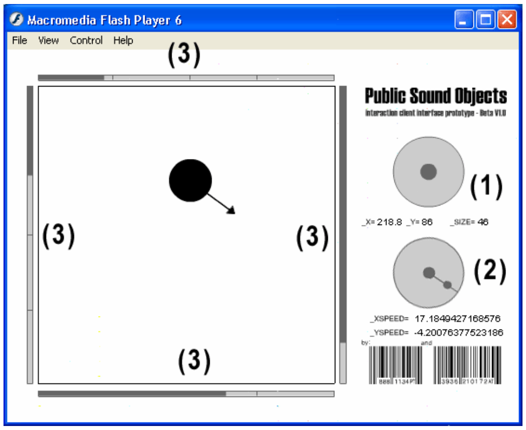
Macromedia Flash Player 6
File View Control Help
(3)
(3)
(3)
(3)
Public Sound Objects
Interaction client interface prototype - Beta V1.0
(1)
_X= 218.8 _Y= 86 _SIZE= 46
(2)
_XSPEED= 17.1849427168576
_YSPEED= -4.20076377523186
by: and

Fig. 3 -The Bouncing Ball Skin

The ball is moving continuously and the user can manipulate its size (1), its speed and direction (2) and each wall's acoustic texture (3).

The normalized values are then sent to the server where they are mapped to the following synthesis parameters:

Modifier #1: The **Wall's acoustic texture** is mapped to the **Sound Object's pitch**. Individual pitch values can be assigned to each wall, allowing the creation of melodic and rhythmic sound structures.

Modifier #2: The **Ball size** corresponds to the Sound Object's **reverberation**. The smaller the ball size, the higher reverberation, following a metaphor of an empty room with a bouncing ball. A bigger ball fills the room and therefore there is less reverberation.

Modifier #3: The **Ball speed** has an influence on the Sound Object's **amplitude**. The bigger the ball, the louder is the sound of the impact in each wall.

The first interface prototype was developed with Macromedia's Flash Action Script language; however the final version will be ported to Java2 mostly because of compatibility and network connectivity reasons.

3.3. The installation site

The installation site for the bouncing ball prototype will be suitable for a media art museum environment, where visitors can either watch the piece, or even participate using one of the provided local clients.

The scenario will be located in a dedicated room, which can hold several people. There will be a video projection showing a single local representation of the bouncing ball interface, visualizing the performance of all current participants.

Various loudspeakers positioned along the walls, create a spatial soundscape reproducing the sounds of the objects colliding with the walls.

4. Conclusions and future work

The Public Sound Objects project is still under development, however, the experiments realized so far with the bouncing ball prototype GUI, and the FMOL synthesis engine, are quite promising.

After this project overview we will finish the prototype implementation, and look for an adequate installation site to guarantee the success of this artistic proposal.

Once the system is operational we will have the opportunity to conduct research and evaluation about the user's behavior and the sonic event's results.

In future implementations we will experiment with the possibility of allowing the users to upload their own Sound Objects to the central server evaluating its musical results. We also intend to explore the possibilities of having different setups adapted to situations with large amounts of simultaneous users. For such scenarios *Micro-Sonic* music techniques [27], the use of banner clients or GRID computing could be interesting approaches.

References

[1] J. Bischoff, R. Gold and J. Horton, *Music for an Interactive Network of Microcomputers*. Computer Music Journal *2,* 24-29 (1978).

[2] C. Latta, *A New Musical Medium: NetJam*. Computer Music Journal *15,* (1991).

[3] D. Gang, V. Chockler, T. Anker, A. Kremer and T. Winkler. *TransMIDI: A System for MIDI Sessions Over the Network Using Transis*. 1997. Proceedings of the International Computer Music Conference (ICMC 1997).

[4] Y. Amir, D. Dolev, S. Kramer and D. Malki. *Transis: A Communication Sub-System for High Availability*. 1992. Proceedings of the 22nd Annual International Symposium on Fault-Tolerant Computing (FTCS).

[5] P. Burk. *Jammin' on the Web: A New Client/Server Architecture for Multi-User Musical Performance*. 2000. Proceedings of the International Computer Music Conference (ICMC 2000).

[6] M. Moller, W. Henshall, T. Bran and C. Becker. *The ResRocket Surfer - Rocket Network*. 1994. http://www.rocketnetwork.com/.

[7] Tonos - online musician's network. *TC8 - Music Collaboration Tool*. 2001. http://www.tonos.com.

[8] J. Young. *Using the Web for Live Interactive Music*. 2001. Proceedings of the International Computer Music Conference (ICMC 2001).

[9] J. Young and I. Fujinaga. *Piano master classes via de Intrenet*. 1999. Proceedings of the International Computer Music Conference (ICMC 1999).

[10] D. Ghezzo, J. Gilbert, A. Smith and S. Jacobson. *The Cassandra Project*. 1996. http://www.nyu.edu/pages/ngc/ipg/cassandra/.

[11] A. Xu, W. Woszczyk, Z. Settel, B. Pennycook, R. Rowe, P. Galenter, J. Bary, M. Geoff, J. Corey and J. Cooperstock. *Real-Time Streaming of Multichannel Audio Data over Internet*. 47[11]. 2000. Proceedings 108th Convention of the Audio Engineering Society.

[12] C. Chafe and G. Niemeyer. *Ping music installation, 2001, SFMOMA*. 2001. http://crossfade.walkerart.org/.

[13] S. Jordà and T. Aguilar. *FMOL: a graphical and net oriented approach to interactive sonic composition and real-time synthesis for low cost computer systems*. 1998. Proceedings of COST G6 Conference on Digital Audio Effects 1998.

[14] S. Jordà and A. Barbosa. *Computer Supported Cooperative Music: Overview of research work and projects at the Audiovisual Institute - UPF*. 2001. Proceedings of MOSART Workshop on Current Research Directions in Computer Music.

[15] S. Jordà, *Faust Music On Line (FMOL): An approach to Real-time Collective*

Composition on the Internet. Leonardo Music Journal *9,* (1999).

[16] M. Mühlhäuser, M. Welzl, J. Borchers and R. Gutkas. *GlobeMusic: The Internet Scale of eMusic-Making*. 2002. Proceedings of the WEDLMUSIC - Web Delivery of Music - conference 2001.

[17] A. Barbosa. *Overview and conclusions of the Music Interfaces Panel Session at the MOSART Workshop (Barcelona, 2001)*. 2001. Report on the MOSART Workshop on Current Research Directions in Computer Music. http://www.abarbosa.org/docs/mosart-interactivity_pannel.pdf.

[18] P. Curtis. *Mudding: Social Phenomena in Text-Based Virtual Realities*. 1992. Proceedings of the 1992 Conference on the Directions and Implications of Advanced Computing.

[19] L. E. Marvin, *Spoof, Spam, Lurk and Lag: the Aesthetics of Text-based Virtual Realities*. Journal of Computer-Mediated Communication *1,* (1995).

[20] W. Duckworth. *Cathedral*. 1997. http://www.monroestreet.com/Cathedral/.

[21] P. Burk. *JSyn - A Real-time Synthesis API for Java*. 1998. Proceedings of the International Computer Music Conference (ICMC 1998).

[22] A. Tanaka. *MP3Q*. 2000. http://fals.ch/Dx/atau/mp3q/.

[23] R. Lozano. *Vectorial Elevation, Relational Architecture #4*. 2000. http://www.alzado.net/.

[24] P. Schaeffer, *Traité des Objets Musicaux.*, 1966.

[25] M. Chion, *Guide des Objets Sonores. Pierre Schaeffer et la Reserche Musicale.*, 1983.

[26] X. Amatriain and P. Herrera. *Transmitting Audio Content as Sound Objects*. 15-6-2002. Proceedings of the AES22 International Conference on Virtual, Synthetic and Entertainment Audio.

[27] C. Roads, *Microsounds*, MIT Press, 2001.