

Interactions in Mobile Sound and Music Computing

Lead Guest Editor: Michele Geronazzo

Guest Editors: Federico Avanzini, Federico Fontana, and Stefania Serafin





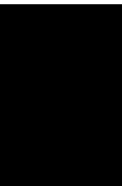
Interactions in Mobile Sound and Music Computing

Wireless Communications and Mobile Computing

Interactions in Mobile Sound and Music Computing

Lead Guest Editor: Michele Geronazzo

Guest Editors: Federico Avanzini, Federico Fontana, and
Stefania Serafin



Copyright © 2019 Hindawi Limited. All rights reserved.

This is a special issue published in “Wireless Communications and Mobile Computing.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Editorial Board

Javier Aguiar, Spain
Ghufran Ahmed, Pakistan
Wessam Ajib, Canada
Muhammad Alam, China
Eva Antonino-Daviu, Spain
Shlomi Arnon, Israel
Leyre Azpilicueta, Mexico
Paolo Barsocchi, Italy
Alessandro Bazzi, Italy
Zdenek Becvar, Czech Republic
Francesco Benedetto, Italy
Olivier Berder, France
Ana M. Bernardos, Spain
Mauro Biagi, Italy
Dario Bruneo, Italy
Zhipeng Cai, USA
Jun Cai, Canada
Claudia Campolo, Italy
Gerardo Canfora, Italy
Rolando Carrasco, United Kingdom
Vicente Casares-Giner, Spain
Luis Castedo, Spain
Ioannis Chatzigiannakis, Italy
Yu Chen, USA
Lin Chen, France
Hui Cheng, United Kingdom
Ernestina Cianca, Italy
Riccardo Colella, Italy
Mario Collotta, Italy
Massimo Condoluci, Sweden
Daniel G. Costa, Brazil
Bernard Cousin, France
Telmo Reis Cunha, Portugal
Laurie Cuthbert, Macau
Donatella Darsena, Italy
Pham Tien Dat, Japan
André L. F. de Almeida, Brazil
Antonio De Domenico, France
Antonio de la Oliva, Spain
Gianluca De Marco, Italy
Luca De Nardis, Italy
Liang Dong, USA



Mohammed El-Hajjar, United Kingdom
Oscar Esparza, Spain
Maria Fazio, Italy
Mauro Femminella, Italy
Manuel Fernandez-Veiga, Spain
Gianluigi Ferrari, Italy
Ilario Filippini, Italy
Jesus Fontecha, Spain
Luca Foschini, Italy
Alexandros G. Fragkiadakis, Greece
Sabrina Gaito, Italy
Óscar García, Spain
Manuel García Sánchez, Spain
L. J. García Villalba, Spain
José A. García-Naya, Spain
Miguel Garcia-Pineda, Spain
Antonio-Javier García-Sánchez, Spain
Piedad Garrido, Spain
Vincent Gauthier, France
Carlo Giannelli, Italy
Carles Gomez, Spain
Juan A. Gómez-Pulido, Spain
Ke Guan, China
Antonio Guerrieri, Italy
Daojing He, China
Paul Honeine, France
Sergio Ilarri, Spain
Antonio Jara, Switzerland
Xiaohong Jiang, Japan
Minho Jo, Republic of Korea
Shigeru Kashihara, Japan
Dimitrios Katsaros, Greece
Minseok Kim, Japan
Mario Kolberg, United Kingdom
Nikos Komninos, United Kingdom
Juan A. L. Riquelme, Spain
Pavlos I. Lazaridis, United Kingdom
Tuan Anh Le, United Kingdom
Xianfu Lei, China
Hoa Le-Minh, United Kingdom
Jaime Lloret, Spain
Miguel López-Benítez, United Kingdom

Martín López-Nores, Spain
Javier D. S. Lorente, Spain
Tony T. Luo, USA
Maode Ma, Singapore
Imadeldin Mahgoub, USA
Pietro Manzoni, Spain
Álvaro Marco, Spain
Gustavo Marfia, Italy
Francisco J. Martinez, Spain
Davide Mattera, Italy
Michael McGuire, Canada
Nathalie Mitton, France
Klaus Moessner, United Kingdom
Antonella Molinaro, Italy
Simone Morosi, Italy
Kumudu S. Munasinghe, Australia
Keivan Navaie, United Kingdom
Thomas Newe, Ireland
Tuan M. Nguyen, Vietnam
Petros Nicopolitidis, Greece
Giovanni Pau, Italy
Rafael Pérez-Jiménez, Spain
Matteo Petracca, Italy
Nada Y. Philip, United Kingdom
Marco Picone, Italy
Daniele Pinchera, Italy
Giuseppe Piro, Italy
Sara Pizzi, Italy
Vicent Pla, Spain
Javier Prieto, Spain
Rüdiger C. Pryss, Germany
Sujan Rajbhandari, United Kingdom
Rajib Rana, Australia
Luca Reggiani, Italy
Daniel G. Reina, Spain
Jose Santa, Spain
Stefano Savazzi, Italy
Hans Schotten, Germany
Patrick Seeling, USA
Muhammad Z. Shakir, United Kingdom
Mohammad Shojafar, Italy
Giovanni Stea, Italy
Enrique Stevens-Navarro, Mexico
Zhou Su, Japan
Ville Syrjälä, Finland
Hwee Pink Tan, Singapore
Pierre-Martin Tardif, Canada



Mauro Tortonesi, Italy
Federico Tramarin, Italy
Reza Monir Vaghefi, USA
Juan F. Valenzuela-Valdés, Spain
Enrico M. Vitucci, Italy
Honggang Wang, USA
Jie Yang, USA
Sherali Zeadally, USA
Jie Zhang, United Kingdom
Meiling Zhu, United Kingdom

Contents

Interactions in Mobile Sound and Music Computing

Michele Geronazzo , Federico Avanzini, Federico Fontana , and Stefania Serafin 
Editorial (2 pages), Article ID 5601609, Volume 2019 (2019)

Creating an Audio Story with Interactive Binaural Rendering in Virtual Reality

Michele Geronazzo , Amalie Rosenkvist, David Sebastian Eriksen, Camilla Kirstine Markmann-Hansen, Jeppe Køhlert, Miicha Valimaa, Mikkel Brogaard Vittrup, and Stefania Serafin 
Research Article (14 pages), Article ID 1463204, Volume 2019 (2019)


A Presence- and Performance-Driven Framework to Investigate Interactive Networked Music Learning Scenarios

Stefano Delle Monache , Luca Comanducci , Michele Buccoli, Massimiliano Zanoni, Augusto Sarti, Enrico Pietrocola, Filippo Berbenni, and Giovanni Cospito
Research Article (20 pages), Article ID 4593853, Volume 2019 (2019)


O2: A Network Protocol for Music Systems

Roger B. Dannenberg 
Research Article (12 pages), Article ID 8424381, Volume 2019 (2019)

The Influence of Coauthorship in the Interpretation of Multimodal Interfaces

Fabio Morreale , Raul Masu, and Antonella De Angeli
Research Article (12 pages), Article ID 9234812, Volume 2019 (2019)

Interaction Topologies in Mobile-Based Situated Networked Music Systems

Benjamin Matuszewski , Norbert Schnell, and Frederic Bevilacqua
Research Article (9 pages), Article ID 9142490, Volume 2019 (2019)

Virtual Net: A Decentralized Architecture for Interaction in Mobile Virtual Worlds

Bingqing Shen  and Jingzhi Guo 
Research Article (24 pages), Article ID 9749187, Volume 2018 (2018)

Editorial

Interactions in Mobile Sound and Music Computing

Michele Geronazzo ¹, Federico Avanzini,² Federico Fontana ³, and Stefania Serafin ¹

¹Department of Architecture, Design, and Media Technology, Aalborg University, Copenhagen 2450, Denmark

²Department of Computer Science, University of Milano, Milano 20133, Italy

³Department of Mathematics, Computer Science and Physics, University of Udine, Udine 33100, Italy

Correspondence should be addressed to Michele Geronazzo; mge@create.aau.dk

Received 7 December 2019; Accepted 7 December 2019; Published 31 December 2019

Copyright © 2019 Michele Geronazzo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Sound and Music Computing (SMC) discipline aims to design better sound objects and environments for promoting multidisciplinary research to understand, model, and improve human interaction in multimodal domains. Moreover, SMC supports core ICT technologies for the ongoing revolution in digital audio and music culture. In particular, mobile and wireless technologies increasingly promote exciting future developments in SMC. Designing of ubiquitous and distributed interactive spaces defines new concepts and challenges of sound control and reproduction. Mobile and embedded input interfaces allow novel control paradigms. Distributed and wearable sensor systems enable the continuous connection and adaptation between mobile sensing technologies and user data (e.g., physiology, gestures, and location information).

This special issue focuses on interactivity in mobile auditory displays, allowing instantaneous sonic/musical feedback as part of action-perception interaction for users. In particular, the low-latency feedback loop between hardware and software is a key element for facing the complexity of spatio-temporal evolution of sound with relevant implications for mobile interfaces between humans and computers. We devote particular attention to the growing maker communities around open embedded hardware platforms that allow the creation of new communication protocols for audio/multimedia data, musical instruments, and interactive audio systems. In particular, we selected six contributions that cover both technical and theoretical aspects of networked communication for shared sound and music interactions. While two of them deal with the intercommunication problem, the next two publications

develop conceptual frameworks for networked music performance in both performative and learning scenarios. The remaining contributions provide interesting insights regarding the interface design process of interactive artifacts and mobile devices considering open platforms.

Roger Dannenberg's *O2: A Network Protocol for Music Systems* hits the interoperability problem between music systems by proposing an extension of the popular Open Sound Control protocol. His contribution puts the accent on the problems of interconnection, unreliable message delivery, and clock synchronization; several computer musicians must deal with as part of their routine activities. *O2* offers solutions to these problems, furthermore making it straightforward for musicians to migrate their distributed music applications to the new protocol, thanks to sharing its roots with Open Sound Control.

A different intercommunication problem is dealt with *Virtual Net: A Decentralized Architecture for Interaction in Mobile Virtual Worlds* by B. Shen and J. Guo. In their contribution, the authors put the accent to the scalability problems posed by such mobile virtual worlds as those sharing interactive music content. These worlds in fact must guarantee high interaction responsiveness also in presence of a large number of users.

Their peer-to-peer solution overcomes mobile device unreliability and communication network instability through a novel infrastructure model, called Virtual Net, providing fault-tolerance in user content management and shared object state consistency.

In *Interaction Topologies in Mobile-Based Situated Networked Music Systems* by B. Matuszewski et al., the

authors present a technical framework to support networked music performance (NMP) and systems, as well as theoretical methodological considerations regarding different aspects of interaction (e.g., social and human-computer). Six case studies with mobile devices in different settings from public installations to concerts and performances are then presented to support such a theoretical framework.

S. D Monache et al. work on a different NMP case study regarding learning scenarios within their paper titled *A Presence- and Performance-Driven Framework to Investigate Interactive Networked Music Learning Scenarios*. The authors detail a conceptual framework for research on a NMP system meant to facilitate shared playing by two musicians in the area of distance and blended learning applications. A preliminary study on chamber music practice meant to explore the effects of latency on presence and quality of the performance in an interactive networked environment.

The Influence of Coauthorship in the Interpretation of Multimodal Interfaces, by F. Morreale et al., addresses the topic of musical interface design from the original perspective of *appropriation* and even *subversion* of interactive systems through multiple coexisting interpretations. The authors introduce a novel design model that can be used to stimulate heterogeneous interpretations of interactive artefacts based on the idea that the design of interpretively flexible systems should embed multiple values and backgrounds at the design stage. The model is illustrated through the case study of Beatfield, a multimodal system, which allows users to control audiovisual material by means of tangible interaction.

Finally, M. Geronazzo et al. propose a portable headphone prototype based on an embedded hardware platform to create an interactive audio story through binaural synthesis. In *Creating an Audio Story with Interactive Binaural Rendering in Virtual Reality*, the design of two simple interactions based on head-tracking and hand controller aims at demonstrating that the quality of the experience could be highly improved compared to regular static audiobooks. A short story based on the horror narrative of Stephen King's Strawberry Springs is adapted and designed in virtual environments in order to evaluate the proposed sonic interactivity.

The selected contributions included in this special issue confirm the interest of the SMC research community in taking advantage of the rapid development of hardware and data connectivity for integrating sound and music interactions into mobile and networked devices. Accordingly, it becomes essential to support research into many novel aspects that are crucial for the development of future mobile interactions within the opportunities offered by acoustic data. The upcoming Internet of Things (IoT), together with the 5G network infrastructure, calls for a paradigm shift especially in the social and human-machine interaction with smart objects. Moreover, ubiquitous computing and artificial intelligence algorithms further foster sensory fusion with sound-related environmental information such as event detection, speech communication, collaborative music making, and many more. Finally, augmented reality technologies require communication networks that are able to

manage massive amounts of data from both real and virtual worlds simultaneously and always respecting real-time constraints which are crucial for interactions with sound.

Conflicts of Interest

The editors declare that they have no conflicts of interest regarding the publication of this special issue.

Michele Geronazzo
 Federico Avanzini
 Federico Fontana
 Stefania Serafin

Research Article

Creating an Audio Story with Interactive Binaural Rendering in Virtual Reality

Michele Geronazzo , **Amalie Rosenkvist**, **David Sebastian Eriksen**,
Camilla Kirstine Markmann-Hansen, **Jeppe Køhlert**, **Miicha Valimaa**,
Mikkel Brogaard Vittrup, and **Stefania Serafin** 

Department of Architecture, Design, and Media Technology, Aalborg University, Copenhagen 2450, Denmark

Correspondence should be addressed to Michele Geronazzo; mge@create.aau.dk

Received 4 January 2019; Revised 21 June 2019; Accepted 4 July 2019; Published 14 November 2019

Academic Editor: Marco Picone

Copyright © 2019 Michele Geronazzo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The process of listening to an audiobook is usually a rather passive act that does not require an active interaction. If spatial interaction is incorporated into a storytelling scenario, can open. Possibilities of a novel experience which allows an active participation might affect the user-experience. The aim of this paper is to create a portable prototype system based on an embedded hardware platform, allowing listeners to get immersed in an interactive audio storytelling experience enhanced by dynamic binaural audio rendering. For the evaluation of the experience, a short story based on the horror narrative of Stephen King's Strawberry Springs is adapted and designed in virtual environments. A comparison among three different listening experiences, namely, (i) monophonic (traditional audio story), (ii) static binaural rendering (state-of-the-art audio story), and (iii) our prototype, is conducted. We discuss the quality of the experience based on usability testing, physiological data, emotional assessments, and questionnaires for immersion and spatial presence. Results identify a clear trend for an increase in immersion with our prototype compared to traditional audiobooks, showing also an emphasis on story-specific emotions, i.e., terror and fear.

1. Introduction

Since 2015, the sale of audiobooks in the United States has expanded by almost 20 percent each year. According to a 2017 survey [1], 24% of Americans (more than 67 million people) have completed at least one audiobook in the last year (2016 data), which resulted in a 22% increase over the previous year (2015 data). There is a promising growth in the field of audiobook storytelling, which calls for a reflection upon the medium and perhaps a look for potential improvements and alterations to be made. A leading distributor and producer in the audiobook field is the Amazon-owned company Audible, which sells and produces spoken audio entertainment, information, and educational programs. However, in all their titles (more than 10000) Audible produces a year, there has been no significant development to the method of storytelling to improve the listening experience. In these traditional audiobooks, there is nothing

more than a narrator or actor reading a story aloud, resulting in a passive experience without nonlinear narratives [2]. Alternatively, the state-of-the-art audio drama created by BBC makes use of static binaural rendering and mixing (listen, for instance, to radio dramas of BBC Radio 3 (<http://www.bbc.co.uk/programmes/articles/29L27gMX0x5YZxkSbHchstD/radio-3-in-binaural-sound>)).

Our main focus is the development of a technological support for audio stories which could benefit from the increasing attention and development of innovative tools in immersive virtual reality (VR). Such a framework could be extended to podcasts and radio broadcasting in general considering mobile VR, e.g., smartphone-required devices such as Google Daydream (<http://vr.google.com/daydream/>) or Samsung Gear VR, and all-in-one devices such as Oculus Go and Quest (<http://www.oculus.com/>).

One relevant aspect able to increase the pleasantness and usefulness of an audio story experience is the level of

immersion. According to Nordahl’s adaptation of Mel Slater’s conceptual framework for describing why individuals may respond realistically when exposed to immersive VR with sound [3], immersion is one of the main four constituents. There are many methods of improving immersion of an audio-only story. Technologies for three-dimensional sound rendering are able to create surrounding sounds, i.e., immersive soundscapes, and interactivity within virtual environments (VE) [4, 5]. Accordingly, the aim of this study is to create a prototype for interactive audio-only stories able to dynamically render spatialised soundscapes with headphones equipped with embedded movement sensors. We manipulated monoaural audio sources in accordance with the object-based audio definition of VE [6], allowing a flexible interaction design and mixing of a narrative in a full three-dimensional space around the listener. Sound propagation and occlusion in VR were kept consistent from the acoustic point of view, providing a rich and natural audio experience. Dynamic spatial audio rendering required head-tracking, allowing listeners to freely move their head localizing and exploring sound sources like in real life [7]. Two micro electromechanical system (MEMS) MPU-9250 sensors able to measure both acceleration and orientation served as the basis for the interaction part of the prototype. After various redesigns based on results from several usability tests, a movement tracker was mounted on noise-canceling headphones. Additionally, we designed a handheld controller for a basic mixing action, i.e., volume balance control between main narrator voice and auditory VE elements.

The proof-of-concept narrative was based on the short story by Stephen King called “Strawberry Springs” and was both shortened and edited in order to be considered in an scientific evaluation procedure. The story’s narrator was then recorded in an anechoic room and played back monophonically, while spatialised sound sources were placed in the soundscape. The progression of the story can be described as a user moving through the storyboard on a predefined path allowing head-movements and volume adjustments. The scene was built in accordance with redesigned storyboard allowing the users to move from the room where the scene was happening, along corridors and scene connections, while triggering sound events or placing sound sources around them (see the Supplementary Material for more details about the storyboard (available here)). A similar approach was successfully adopted in [8] with the idea of *music rooms* for enhancing music genre learning with minimal visual feedback. A first evaluation was conducted comparing our final prototype with two other experiences: a passive monophonic listening (traditional audiobooks) and a static binaural listening (state-of-the art audio storytelling).

2. Background

2.1. Interactive Storytelling. Storytelling can be described as “the activity of telling or writing stories” (from Oxford Learner’s Dictionary, <http://www.oxfordlearnersdictionaries.com/definition/english/storytelling>). It is a social and cultural act, which can include theatrics, improvisation, and the likes. As mentioned by [9], storytelling has been around for a long

time and serves not only to recall past events but also to spread awe by way of fantasy, fiction, and “magic.” While modern civilization continues the age-old tradition of telling stories both orally and visually, it also finds more innovative ways of telling them.

Interactive storytelling is a fairly new field from the late 1980s and 1990s, related to many different fields, like games, cinema, storytelling, programming, and mathematics [10]. When combining storytelling and interactivity in order to achieve interactive storytelling, it is important to maintain balance between the amount of control the user has over the story and the coherency of the story [11]. When considering what is required to create an interactive experience, the design of the interaction and the feedback are tightly connected [12]. Users need the ability to influence the feedback they receive before a product can be considered interactive. Influence and feedback can be achieved in several different ways. Before the digitalisation of stories, some of the ways of creating interactive stories was during public readings [13]. The reader or *teller* would actively engage the audience to prompt a response, which would either alter or further immerse the audience in the story based on the response. One can also describe this process as forward leaning or participatory storytelling, as opposed to conventional backward leaning or hypnotic storytelling achieved by a noninteractive experience.

2.2. Binaural Audio Technologies. Understanding how humans process everyday sounds is extremely relevant when designing an audio system. Simultaneous sound events in the environment can be identified, and it is possible to focus on a specific sound. The auditory system can detect the origin of a sound, elaborating its direction-of-arrival (DOA) through the head-related transfer function (HRTF), and changes in interaural level and time differences. In particular, HRTFs describe the acoustic characterization of the human body for point source around the listener, being highly individual especially for vertical sound localization [14, 15] Interestingly, sounds that are memorized through repetition are more easily identified by humans [16].

Binaural techniques usually refer to methods for recording and reproducing sound with the intent to construct an immersive auditory sensation. A common method for this is the use of so-called dummy head recording, which involves two microphones placed at blocked ear canal position of artificial head’s ears, with their two isolated outputs being played to a pair of headphones worn by the listener [17]. Since the dummy’s ears are built to resemble a real human ear, the sound waves are modified during the recording process, and approximate how the sound waves would be altered in a real-life scenario before reaching listener eardrums. Even though binaural recordings are not widespread within the music industry, they are being used for ambient experimental music and sound stories [18]. Furthermore, more flexible rendering techniques for sound propagation are being developed, providing sophisticated sound engine software especially for games, such as Google Resonance (<https://github.com/resonance-audio>), Steam

Audio (<https://valvesoftware.github.io/steam-audio/>), and Wwise (<https://www.audiokinetic.com/products/wwise/>), to name but a few.

3. Related Work

3.1. Spatial Orientation with Sound. Head and body movements are important for building cognitive maps of the real/virtual space around the listener, especially for visually impaired people [19]. Binaural audio technologies play an important role in conveying relevant spatial information via headphones in order to acquire the integrated knowledge for spatial orientation and navigation [20]. Spatial orientation also refers to how it is possible to track a user's location and orientation for rendering purposes. One can identify three common methods to manipulate user orientation in VEs: head-, body-, and device-tracking. In [21], authors tried to find differences among tracking methods when the user was moving toward a sound source in an audio-augmented reality scenario. Their results showed that there were not any statistically significance differences between the three tracking methods in terms of localization performances. On the other hand, if time to accomplish the navigation task and realism were considered, head-tracking would be the best solution. Regarding technical requirements for a head-tracking implementation, Hess [22] argued that the latency from head movement to virtual feedback can be maximum 62 ms.

Finally, it is worthwhile to notice that virtual room acoustics is important for a natural perception of reconstructed sound scenes, providing a recognizable acoustic fingerprint of specific location or event [23]. This is particularly relevant for echolocation abilities, which rely on DOA of echos in the room which could be effectively rendered with current VR technologies [24], such as those employed in this study (see Section 2.2 on binaural audio technologies). Moreover, thanks to the circular interaction between spatial presence and emotions: one can consider VR an affective medium [25] which is able to interact with user's affective states [26] and memory processes [27].

3.2. Interactive Audio Stories. In the scientific literature, there are many attempts in introducing interactivity in the passive listening of audio stories. Furini [2] developed a system architecture able to turn listener into the story director. The author proposed a script manager able to support a producer in the definition of atomic audio scenes/segments with meta data in MPEG7-DDL language and their allowed connections, i.e., a story graph. Listeners used the interaction manager in order to look at the scene transition table for possibilities in the story path. The user interface could be implemented within a touch screen or a voice detection system. In [28], Huber et al. focused on the user interface design, story sonification, and game interaction of nonlinear narrations. They defined interactive and narrative *nodes* in the story, with the opportunity to create nodes with mini-games and static 3D audio objects based on OpenAL rendering capabilities (<http://www.openal.org>). More recently,

Marchetti and Valente [29] explored the untapped potential of audio in the context of a foreign language learning in primary and secondary schools. They were developing a prototype of mobile application which offered a multimodal experience in extending reading with social interaction: a platform for annotating, tagging, and sharing comments from written book to the correspondent audiobook.

It is worthwhile to notice that music also plays an important role to elicit the proper emotional response during a story. For example, automatically generated music scores might increase the overall quality with respect to audio stories without music [30].

Finally, a wider prospective of the use of audio narration could be in the automatic creation of daily stories based on user data from ubiquitous and wearable technologies (e.g., mobile devices/sensors) [31]. Such data presentation might help users in navigating effectively the increasingly amount of personal information in order to gain self-awareness or produce a desirable behavioral change.

3.2.1. Commercial Applications. *Koob* virtual reality audiobooks (<https://www.koobaudio.com/>) was created by the Voice Society and is a digital platform. KOOB wanted to revolutionize storytelling by using VR sound to create life-like immersive experiences in order to drag the listeners mind deep into the story, making them feel like they are living the story themselves. This was done by tricking the listeners mind into thinking that it was experiencing the binaural sound with intuition, creativity, and imagination, understanding, and reasoning.

The Owl Field (<https://www.owlfield.com/>) is a company which creates immersive binaural audio dramas. Their audiobooks are told from a first-person perspective where the events, characters, sound effects, and music surround the user. An interesting aspect to *The Owl Field's* storytelling is that the listener is the story's central character. The characters of the story speaks to the listener, and involves them in the story. The use of binaural audio recordings is what sets *The Owl Field* aside from common narrated audiobooks. Additionally, making the user an integrated part of the story gives potential for high immersiveness.

Hellblade: Senua's Sacrifice is a dark action-adventure game developed and published by the studio Ninja Theory in August 2017. The game was set in a fantasy world build on Norse and Celtic mythology, and followed the young warrior, Senua, and her journey to Hel, where she hoped to save the soul of her murdered lover, Dillion (read <https://www.giantbomb.com/reviews/hellblade-senuas-sacrifice-review/1900-765/> for a review). The caveat was that Senua suffered from psychosis, meaning that she suffered from hallucinations, and as such, the world you visit was a manifestation of her mind. Furthermore, it resulted in schizophrenia and auditory hallucinations, i.e., Senua hearing voices. The general consensus between games journalists and reviewers was that the game's sound design based on binaural recordings was able to build great immersion by replicating aspects of psychosis and was a great narrative accomplishment.

4. The Interaction Design in Our Narrative

To make the user feel present in the scene, one could argue that the aspect of immersion related to spatial presence can be pursued by implementing a spatialised soundscape where the digital sound sources respond to physical movement like they would in real life. This can be achieved by implementing a dynamic audio environment played through a head-tracking pair of stereo headphones. In relation to implementing a more immersive audio experience, spatial sounds can be designed to enhance a sense of presence like in many video games [32]. Although real-time audio-processing can be computationally expensive, GPU acceleration and the fact that modern computers have become very fast helps reduce this problem [33]. Additionally, since such a kind of system does not include any graphics in the prototype, practically all computing power will be available for audio-processing.

4.1. The Narrative. Stephen King’s Strawberry Spring story was written in 1978; however, it is mostly set in a flashback in 1968. In the following, we give a brief summary of the story: it revolves around a springtime murder spree that happened at the narrator’s college in 1968. The weather is referred to a phenomenon known as “strawberry spring” in the story, due to the heavy fog. The narrator seems less affected than others around him and acquires a fascination for the murderer. Ten years later, in 1978, after the murders have been forgotten, the fog reappears and the narrator fears he has committed the same crimes as many years earlier.

The narrative was arbitrarily shortened and adapted for this interactive audio medium and its evaluation, trying to identify peculiar aspects of such design process. Furthermore, some words were changed in order to better adapt to the current time period, rather than the 1960s.

This specific story was chosen, based on the genre, duration, and first-person narration. In particular, this latter aspect allows the characters to directly interact with the user. Furthermore, this peculiar aspect demands the user to be active in listening, and cannot passively walk away from the narration. Spatial sound in the soundscape, such as cafeteria background sound and quiet conversations, and sound effects, such as wolf howls and crickets chirping, were actively placed around the users while they proceed in the time-line of the story. Moreover, the story segments/events and their audio objects were created within virtual rooms with virtual walls, corridors, and open spaces to take advantage of spacial interaction in room acoustics, e.g., sound occlusions, as design parameter (see Section 4.2 for further details). Figure 1(a) depicts the top view of the implemented story map made of such virtual elements connected in order to create the desired storyline. The adopted narrative flow was arbitrarily designed keeping in mind a feasible and short experimental evaluation for the listener (see Supplementary Material for time-stamps and durations of narration segments in the storyboard allowing a tangible time-line for testing and reproducibility).

4.2. Sonic Interactions. The listener was forced to move along a predefined track in the audio-only VE, which spatially described the narrative. The track went through several virtual rooms and corridors that acoustically resembled those described in the narrative, as this provided a realistic representation for the spatial sounds. For such purposes, the audio engine provided a set of materials which have their own unique acoustic properties, e.g., occlusion, sound absorption, and reflectance. Moreover, each recording was numbered and placed in chronological order, and each story segment was placed to take the user through the narrative in the correct order. The track system was the primary method for forcing the navigation into the story at specific movement speed, allowing the increase or decrease of speed with which the user would visit each segment/place, and thereby the speed of the story.

The map was built in accordance with the storyboard. Though the user did not see the visual counterpart, the story was spatially mapped visually for faster editing of source placement and organization by the experimenters.

4.2.1. Static and Traversal Scenes. The segments in the storyboard were given a “Static” or “Traversal” tag depending on whether the narrative prompted a virtual scene in a closed space/room or a scene with movements (see Figure 1(b) for an example of such distinction). The changes in speed most often occurred when users went through a change between a static and a traversal scene.

Enclosed virtual rooms correspond to static scenes, in which the user’s position (not orientation) is fixed. This is meant to give a sense of being immersed in the corresponding soundscape. Static scenes were prompted by narrative phrases and conversations, with a low-speed position change. The traversal scenes were spatially described by long winding paths, through which the user would traverse through the corresponding soundscape. Accordingly, traversal scenes were commonly associated with high-speed position change.

4.2.2. Soundscape. As the user passed through the virtual scene, sounds were triggered around them. The sound samples are grouped into four categories: narrative (48 samples), ambience (25 samples), movement (6 samples), and other sounds (40 samples), some of which also were considered *reaction sounds* for special events in the story (see Supplementary Material where each category has their own colored tag for a quick overview of the complexity of the resulting short story).

The sounds were played by adding a trigger collider to the source and a box collider to the listener position in the story. Figure 1(c) depicts an example where the green sphere is the trigger radius or area of a sound source. The narration clips were separate to the ambient clips and were also triggered by collisions. The difference is that the sound source playing the narration clip was attached to the user and located slightly behind the audio-listener object. This created the effect of the sound being played behind the user. An example of such interaction could be extrapolated from

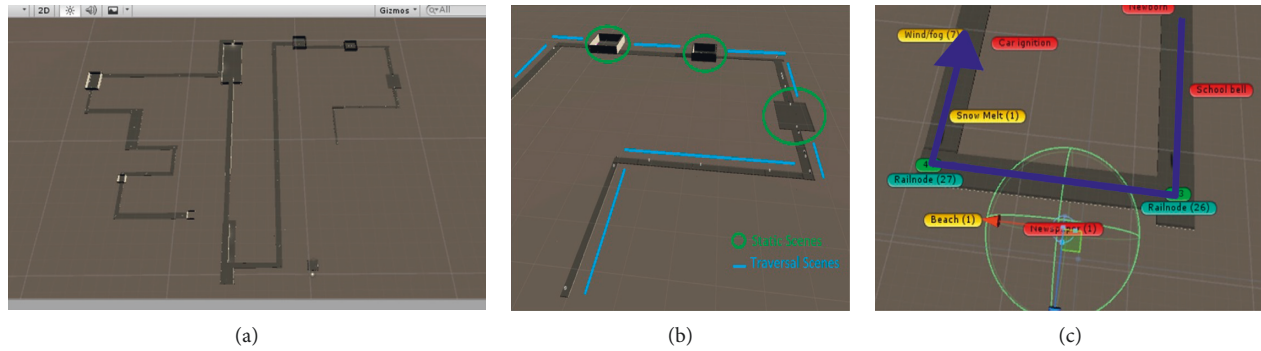


FIGURE 1: (a) Visual mapping of the story in VR; (b) mapping of static and traversal scenes; (c) a section of the story where the user path (blue arrow) triggers *ambient sound* (yellow labels), *narrative* (green labels), and *sample audio* (red labels).

Figure 1(c) where the traversal scenes were prompted by narrative phrases such as “Then, today’s paper” and the sound of turning pages was triggered in the interactive story.

4.2.3. Additional Controls. A study by McGloin, Farrar, and Kramer aimed to investigate whether or not the controller matters when it comes to immersion in video games [34]. Their research suggests that controllers are in fact capable of influencing the perceived realism. They present the concept of controller naturalness, which refers to the overall intuitiveness a controller is perceived to have when interacting with a VE [35]. Accordingly, a remote controller was initially designed in order to provide the following basic functions:

- (i) Pausing the narrative, allowing the user to explore the soundscapes
- (ii) Adjusting the relative volume of the soundscape with respect to the narrative

This latter functionality should follow the gesture of turning a knob while adjusting the volume on a speaker: turning it left, or counterclockwise will turn down the volume, and turning it right, or clockwise, will turn it up. It is worthwhile to note that the pausing action was implemented but not considered in the final prototype according to a first usability test briefly described in Section 5.2.

5. Prototype Implementation

The creation of the story and its VE was based on Unity framework (<https://unity3d.com/>) which is an extensive game/application development kit. We used the capabilities and parameters of the Steam Audio engine (<https://valvesoftware.github.io/steam-audio/>), which enables a real-time rendering of realistic VE with specific audio parameters, such as audio occlusion, reflection, and propagation. Our system employed default Steam Audio rendering parameters (e.g., generic HRTFs). The narrator’s voice was recorded in the anechoic room of the Multisensory Experience Lab (<https://melcp.create.aau.dk/>) at Aalborg University Copenhagen, with a Zoom H6 sound recorder. Such recordings composed a monophonic track resulted in *in-head* localization of the voice for the users, which guided them through the story without

changing position in space. Sound effects and soundscape were selected within various collaborative databases of creative-commons licensed sounds, e.g., Freesound (<https://freesound.org/>).

Figure 2 captures the two main components of the prototype. In Figure 2(a), one can see the head tracker that was secured in a small wooden case. In Figure 2(b), a shortened white PVC pipe contained the movement sensor and the button. Arduino Nano 2.3 was employed for a fast prototype of the devices. Specifically, two 9-axis gyroscope MPU-9250 sensors were connected to the Arduino board, which interacted with the Arduino through a I2C bus. The Arduino had both SDA and SCL pins and the platform came with built-in support functions, which made easy to implement the I2C protocol. Only the outputted angular-velocity values (Gy^x, Gy^y, Gy^z) were used in the computation of user’s movements. Arduino was further connected to the computer by interfacing the Arduino board with a USB cable. Data were acquired in the Arduino IDE and sent to Unity through a serial port.

Finally, a custom range (−20 dB and +15 dB full scale) was set for the volume control of the soundscape in order to isolate the narrator’s voice, and the lower bound effectively silences the immersive soundscape. The button might pause the linear movement in the narrative; while rotating the device horizontally, the user can adjust the volume of the ambient noises.

5.1. Remote Controller Connection. The bitrate for the serial communication between Arduino and Unity was set to 9.6 kbit/s. In the Unity environment, after instantiating and opening the serial port, the data logger was started. If an exception occurred during initialization, the script was disabled. This ensures that the program can still run without a connected Arduino, which is useful for development purposes.

An *update* method was called every frame, allowing the reading of sensors data from the Arduino. The processing steps could be summarized as follows:

- (1) Reading incoming sensor data and assigning the corresponding variable

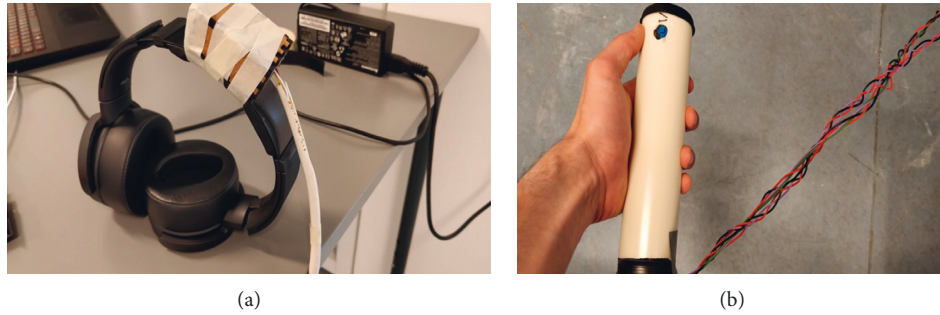


FIGURE 2: Main components of our prototype: (a) noise-canceling headphones equipped with head tracker; (b) gestural handheld controller (remote).

- (2) Parsing the raw string data to a floating-point value that can be used in Unity; a normalization factor was also applied before sending the value
- (3) Comparing the value with a noise-threshold, thus not considering very small sensor responses
- (4) Applying a gain to the variable controlling the volume
- (5) Parsing the button state, i.e., its Boolean value
- (6) Sending a pause signal to user movement in the Unity VE if the button press is true
- (7) En-queuing the button state and volume for data-logging in a separate thread

The volume value was then applied to the audio-mixer of spatial sounds in the soundscape.

5.2. Usability Test and Final Prototype. The test was conducted in an ad hoc Unity scene where the user moved forward automatically, while being exposed to spatialised sound from the surroundings.

Five users were asked to perform three actions in three different navigation trials:

- (1) Moving their head using the head-tracked headphones
- (2) Moving the remote controller horizontally
- (3) Using the button on the remote

After each trial, users were asked to explain what they thought happened with the specific action.

Table 1 showed that users interacted with the headset intuitively and in some extent also with the remote. However, some issues with the button were identified. Two out of five participants understood the concept of the button and described it as the sound in the surroundings being paused momentarily. The other three participants, when questioned about the button on the remote, replied with either confusion of that they felt no effect of clicking it, which might be because there was no immediate feedback when pressing it. When asked about the effect of the horizontal movement of the remote, almost all test participants formulated that it controlled the sound volume.

It could be concluded that the head tracker and remote were both working efficiently enough to go into final testing. The

button still lacked a reliable functionality and had to be redesigned again or removed entirely. As a consequence, the button functionality was remapped for activating the volume control through gesture: when the button was held down, the user was able to direct the remote in either direction horizontally and change the volume for the spatial sounds. Left direction will turn down the volume, and the right direction will turn it up.

6. Evaluation

The primary goal of this study is to explore if interactions with auditory VR contents create a more immersive experience for the user in audiobooks. This means that the prototype has two distinct features: binaural audio rendering and sonic interactions. These two components should be tested separately in order to identify to which extent they impact the feeling of immersion, thus leading to three experimental conditions in a between-subject experimental design:

- (1) Monophonic playback without interactions—common technology in audiobook (MA hereafter)
- (2) Binaural audio stereo playback without interactions—state-of-the-art solutions in audio drama/book (BA hereafter)
- (3) Binaural audio rendering with interactions—our proposed solution (IVE hereafter)

The experimental conditions will be tested on separate sample populations in order to maintain the novelty of experiencing the audio narrative for all participants.

However, this approach introduces a couple of issues. Firstly, there could be individual differences in the participants in each sample group in terms of personality, temperament, etc. To counter this issue, samples of similar age and ethnographic background should be used in the test and the participants were randomly assigned to an experimental condition. Thirty participants (ages M:22 SD:5) took part at our evaluation stage, all were student volunteers, self-reporting, normal hearing, and with no past experience on binaural audio.

6.1. Protocol. The evaluation was conducted in three sessions, one for each of the conditions, with 10 different

TABLE 1: Qualitative data from the usability testing.

Sub ID	Headset	Remote	Button
1	It switches sounds but in an unknown way. It rotates your head around in the sound space	Intensity/volume of the sound	No understanding.
2	The sound moves with the direction that I am looking. It felt like I was able to explore different scenarios. Felt natural when looking around	Adjustment of intensity/focus of the sound	Seems like it saves the sound when you click it. Heard the sound for a longer period of time.
3	When I turn my head, it is like the sound moves with me. Sound becomes more clear depending on the direction. Much like in real life	Move left the sound becomes low and muffled. Move right sound becomes louder and clearer.	I did not feel much happen.
4	It is like normal audio navigation. Able to locate sound source according to head movement	Quite intuitive volume control.	A single sound continued playing. Like the other sounds stopped moving past me.
5	Sounds are passing by and I can follow them with my headset. Feel like naturally moving through sound	Did something to the intensity of the sound but not sure how it works.	I felt no difference.

participants per condition leading to a between-subject experimental design. The experimental procedure was conducted in accordance with the Declaration of Helsinki (Edition 2013).

External uncontrolled stimuli were removed as much as possible. Test participants were placed in a dark room, ensuring that they were not influenced by external sources of light or sound that could confuse them or cause a loss of attention.

The participants were asked to sit down right away, in order to get their heart rate lowered. The participants were then introduced to the project and told about the test they were about to participate. For all of the conditions, the participants had to wear the E4 wristband from Empatica throughout the test, in order to track their heart rate. The wristband was placed on the participants' left wrist.

In conditions without interactions (MA), the participants were simply asked to wear the headphones and listen to the story. For the condition with interactions (IVE), the participants were asked to wear the headphones with the head tracker attached and hold the remote object in their dominant hand, after which they were told about the three kinds of interactions which were possible while listening to the story.

6.2. Metrics. The purpose of the proposed evaluation is to measure state immersion in a fear-inducing experience.

Immersion is a difficult term to describe. It can be divided in two separate terms; flow and spatial presence [36]. Flow is the psychological state of absorption and extreme concentration on a task at hand [37], whereas spatial presence refers to feeling physically present in a mediated environment. Measuring immersion is a challenge. Most studies use questionnaires to quantify it, but this method is sensitive to the user's opinion, mood, and other external factors that the experiment setup cannot control [38]. In an attempt to counterbalance this, a mixed-methods approach was used, as this method was effective in ensuring a broad and deep understanding of the user:

- (i) Monitoring the heart rate of the user
- (ii) Providing a questionnaire to assess the user's self-assessment

In particular, a physiological measurement is indeed helpful in reading the emotional state of the user, but it is unreliable in determining which type of emotion is being experienced [39]. Therefore, the physiological measurement was only used to support the self-reported data.

The nature of the project allows for a great degree of creative freedom, so the type of emotional response the prototype intends to trigger can be adapted according to what is easier to measure. The narrative of the prototype is bound to have certain themes, or a genre, that aim to produce a specific emotional response in the listener. We chose a target emotion that can be effectively measured; fear is one of many emotions looked at, and also an appropriate response in the horror genre.

Fear is the emotion associated with pain, danger, harm, etc. It bears close resemblance with anxiety, and, according to Freud (in 1924), there is a direct link between the two, and he considered the two to be the same emotion in many ways [40]. Furthermore, Freud in 1936 linked anxiety to exploratory behavior, indicating that anxiety leads to curiosity.

In this context, an important feature of fear is that the test participants were unlikely to be biased by this emotion when starting the experiment, given that they were not aware of the potentially fear-inducing experience they were about to participate. Oppositely, if happiness or sadness was measured, the participant would be biased about this emotion according to how their day was going and other trivial aspects of their life that could influence these emotions [38].

6.2.1. Immersive Response. In [41], the authors developed a quantitative measure for immersion in video games called the IMX questionnaire. Our aim is not about creating a video game, but rather an interactive story, but the two are

somehow similar that the items in the questionnaire are very relevant for us, with a few exceptions. The original IMX questionnaire consists of three parts; one that applies to all games, one that applies to games with characters, and one that applies to multiplayer games [42]. For the sake of simplicity, only the first part was used in this study, as the two others were less relevant to our purposes. Starting from them, eight questions were adapted to the context of narrative and are listed below:

Q1: The sound experience felt consistent with the real-world experience

Q2: After playing, it took me a moment to recall where I really was

Q3: During the experience, I felt at least one of the following: breathlessness, faster breathing, faster heart rate, tingling in my fingers, a fight-or-flight response

Q4: The story stimulated my reactions (panic, tension, relaxation, suspense, danger, and urgency)

Q5: Having to keep up with the speed of the story pulled me in

Q6: The story was energetic, active, and there was a sensation of movement

Q7: The story was thought-provoking for me

Q8: I felt caught up in the flow of the story

The responses were collected in 5-point Likert scale.

6.2.2. State Emotions. In the scientific literature, several theories of discrete emotions have been proposed [43]. Specific sets of discrete emotions can be extracted from research on facial or behavioral expressions, and from direct brain stimulation of animals.

In a story, multiple state emotions could be solicited. In order to measure the self-reported state emotions, the Discrete Emotions Questionnaire (DEQ) was used [44]. The questionnaire asks the participant to rate a number of emotions such as anger, dread, terror, scared, and fear using a 7-point Likert scale, according to how they felt during the experience.

6.2.3. Heart Rate Data Collection. Heart rate and pulse are two separate measurements, yet they are closely related, since the pulse stems from the heart. The heart rate is the number of heartbeats occurring in one minute. The average healthy adult heart rate is 60 to 80 bpm (beats per minute), where the normal for older adults is considered to be 60 to 100 bpm, and is affected by several conditions, e.g., emotional state, exercise, and stress [45].

The optimal method for recording heart rate usually employs ECG (electrocardiography), which is the process of recording electrical activity generated by the heart by placing electrodes on the skin. However, the Empatica E4 wristband was used for monitoring the heart rate in a more practical way for the participants.

A study on the accuracy and reliability of four different commercial heart rate monitors shows a varying accuracy

when compared with electrocardiograms. As such, the wrist-worn monitors showed generally better accuracy during resting and declining with exercise [46]. Additionally, a similar study, where one of the two employed heart rate monitors is Empatica E4, showed that accuracy is once again very dependent on level and amount of movement during monitoring. The accuracy of the measured heart rate is evaluated with respect to electrocardiography, and the absolute difference of the measured heart rate and electrocardiograms was less than 10 bpm for 81–97% of the time for E4, while the percentage of accurately detected heartbeats was 68% during sitting, but only 9% during household work. The study concludes that wrist-worn devices such as the E4 are accurate and reliable for heart rate detection when hand movement is not excessive [47].

The use of a wrist-worn monitoring device, specifically the Empatica E4 at 64 Hz sample rate, is deemed sufficient for this study, since the participants are supposed to be sitting down with only moderate movement.

7. Results

The data statistics from the IMX questionnaire are plotted in Figure 3. The responses were nonnormally distributed, so a nonparametric one-way Kruskal–Wallis ANOVA test was used. In the following, results from each questionnaire items are reported: Q1 ($\chi^2(2) = 6.64$, $p < 0.05$), Q2 ($\chi^2(2) = 7.92$, $p < 0.05$), Q3 ($\chi^2(2) = 3.72$, $p = 0.15$), Q4 ($\chi^2(2) = 2.41$, $p = 0.30$), Q5 ($\chi^2(2) = 1.48$, $p = 0.47$), Q6 ($\chi^2(2) = 0.86$, $p = 0.65$), Q7 ($\chi^2(2) = 0.84$, $p = 0.65$), and Q8 ($\chi^2(2) = 0.85$, $p = 0.65$).

To find out which particular conditions differed, a Mann–Whitney test was conducted. Pairwise comparisons for statistically significant results are shown in Table 2. Questionnaire items 1 (sound consistency with reality) and 2 (spatial presence) showed differences between conditions MA and IVE; item 2 also showed a difference between condition BA and IVE while item 1 showed a value close to $\alpha = 0.05$ between condition BA and IVE. These results suggested a differentiation of IVE towards a more immersive experience.

The nonparametric one-way Kruskal–Wallis ANOVA test is used to test for significant differences between the conditions in DEQ for each emotion in the questionnaire: anger ($\chi^2(2) = 1.21$, $p = 0.49$), sad ($\chi^2(2) = 0.3$, $p = 0.86$), grossed out ($\chi^2(2) = .19$, $p = 0.91$), happy ($\chi^2(2) = 3.0$, $p = 0.22$), terror ($\chi^2(2) = 5.8$, $p = 0.05$), rage ($\chi^2(2) = 1.8$, $p = 0.41$), grief ($\chi^2(2) = .06$, $p = 0.97$), anxiety ($\chi^2(2) = 8.4$, $p < 0.05$), nervous ($\chi^2(2) = 4.8$, $p = 0.09$), scared ($\chi^2(2) = 3.8$, $p = 0.15$), sickened ($\chi^2(2) = 0.19$, $p = 0.91$), fear ($\chi^2(2) = 6.7$, $p < 0.05$), calm ($\chi^2(2) = 2.5$, $p = 0.28$), and panic ($\chi^2(2) = 10.2$, $p < 0.01$). In order to get a deeper insight into how each condition affects the emotions, each combination of conditions was subjected to a Mann–Whitney test. The resulting significant p values are shown in Table 3, showing that the emotions terror, anxiety, nervous, fear, and panic have statistically significant differences between MA and IVE. Additionally, a difference is also detected for terror between conditions MA and BA. Moreover, Figure 4 shows a pie chart of this relevant emotions, for

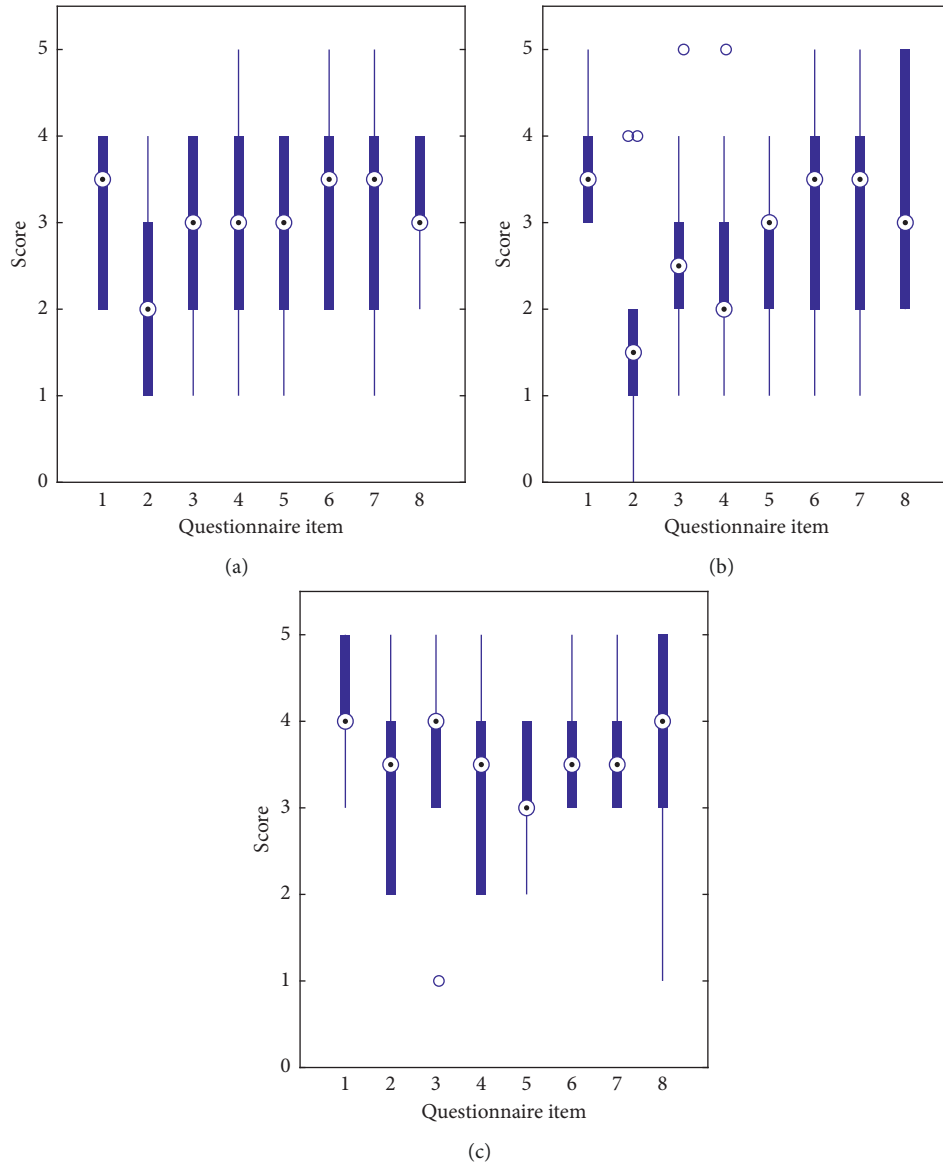


FIGURE 3: Boxplots of all items of the proposed IMX questionnaire grouped by listening condition. (a) Monophonic—static listening (cond. MA); (b) binaural recordings—static listening (cond. BA); (c) interactive virtual environment (cond. IVE).

TABLE 2: p values of pairwise comparisons in IMX questions with meaningful statistical effect.

Cond. comparison	p values	
	q1	q2
MA vs. BA	0.44	0.45
MA vs. IVE	0.02	0.03
BA vs. IVE	0.06	0.02

TABLE 3: p values of the pairwise comparison in emotions rating.

Cond. comparison	p values			
	Terror	Anxiety	Fear	Panic
MA vs. BA	0.68	0.02	0.38	0.17
MA vs. IVE	0.03	0.01	0.01	0.00
BA vs. IVE	0.07	0.91	0.12	0.10

each of the three test conditions. These results suggested a clear trend for an emotional enhancement for IVE experience with respect to conditions MA and BA.

From the raw data, the average heart rate and the deviation from that average were computed for each participant (see Figure 5 for a boxplot representation). The three

conditions were evaluated with a one-way Kruskal–Wallis ANOVA which led to no statistically significant differences among conditions in average ($\chi^2(2) = 2.3$, $p = 0.31$), and standard deviation ($\chi^2(2) = 0.67$, $p = 0.71$). Moreover, the ratio between the first and last minute of narration was computed in order to identify a persistent increase in heart rate due to the experience. Again, no statistically significant

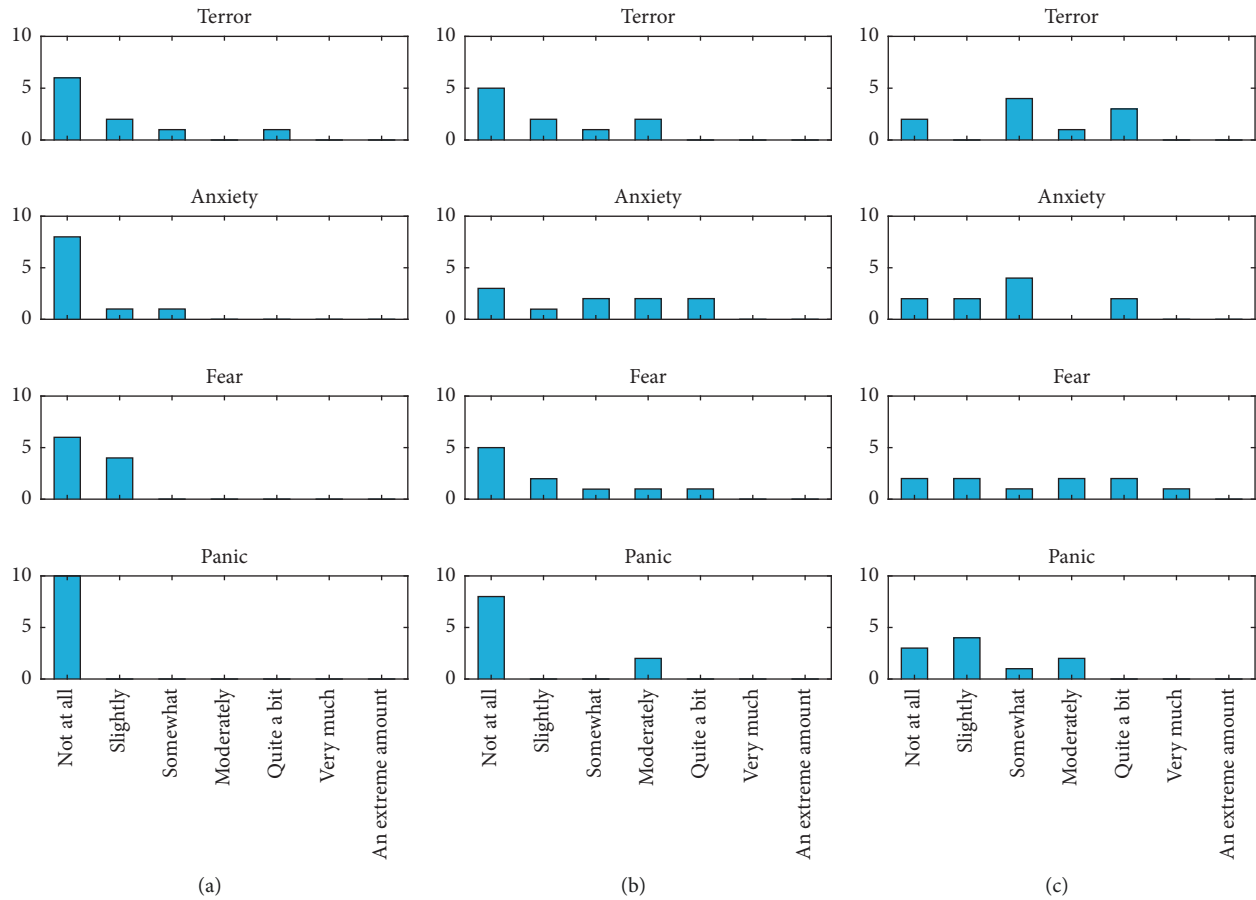


FIGURE 4: Data distribution of rates for the relevant emotions/items from DEQ questionnaire grouped by experimental condition. (a) MA; (b) BA; (c) IVE.

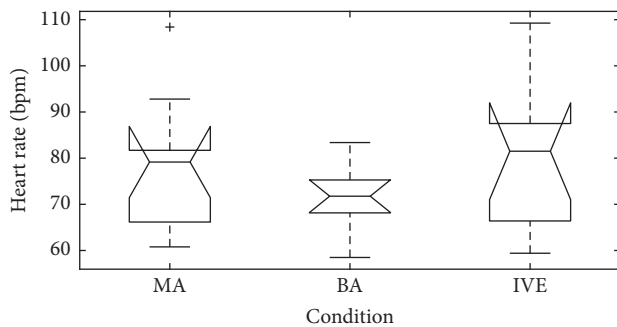


FIGURE 5: Notched boxplot representation of heart-rate data distributions for each condition.

differences were found among conditions ($\chi^2(2) = 1.6, p = 0.6$). However, the average trend suggested a higher heart rate together with a higher variability among users for the IVE experience.

8. General Discussion

By considering the IMX questionnaire with the DEQ, both the level of fear and immersion were measured, and it was interesting to evaluate these two aspects in the context of a horror-experience.

Our results showed that there was an interaction between spatial sound rendering and interactivity that enhanced the feelings of immersion with fear-induced emotions. Since analysis on heart rate data revealed no statistical difference, the questionnaires suggested that there was a difference in immersion and emotions between monophonic narration (MA) and immersive virtual environment (IVE), which was not always visible in the comparison between monophonic narration and static binaural recordings (BA). Moreover, the collected data suggested a trend of increasing level of immersion and fear-related emotions related to interactivity.

It has to be mentioned that there were several confounding variables that could have affected the heart rate data. Mainly, there was little control over the participant's initial heart rate, so a baseline resting heart rate could not be established. Instead, the average heart rate was used as the baseline, and the average deviation from that baseline was used as the main statistic. The goal was to measure how much the heart rate varied throughout the narrative. However, the development of the heart rate through the narrative could not necessarily be attributed to immersion in the narrative. If a participant had been physically active before the experiment, they would have an elevated heart rate at the beginning of the experiment, and it would decrease during the narrative as a

result of the participant not moving. To mitigate this effect, the participants could have been asked to sit still for half an hour before the narrative and then establish a baseline resting heart rate after this period of rest.

In the IMX questionnaire, items Q1 (The sound experience felt consistent with the real-world experience) and Q2 (After playing, it took me a moment to recall where I really was) showed significant differences. When looking at averages for each test condition in Figure 3, there seems to be a small, but noticeable, increase for condition IVE. Overall, all IMX answers show an increase in immersion between conditions MA and IVE on average, not detectable between MA and BA.

The first item from the IMX questionnaire is closely linked to the immersion concept of spatial presence, so detecting a difference in this question is an indicator that the level of immersion was increased. This question indicates that IVE condition provided a more realistic sound experience, and this could indicate that it was easier for the user to reach a state of spatial presence under such a condition. However, more data have to be collected before this claim can be fully asserted. The second IMX item is also an indication of increased immersion. The question can be linked to the concept of *flow*: cause a loss of self-consciousness and alter one's sense of time. This is another indicator that the experienced immersion varied with the conditions.

It is worthwhile to notice that the static binaural audio rendering (BA) led to a similar (sometimes in a worse trend) answer in IMX to MA which seems counterintuitive considering the big difference between monophony and binaural audio. However, Steam Audio engine employed generic dummy-head HRTFs which are known to cause frequent front-back confusion and inside-the-head localization without head-tracking and because of their generic solution they are not good for every listener [15, 48]. The discrete emotions questionnaire supported the result of the IMX questionnaire. The ANOVA test revealed that the emotions terror, fear, anxiety and panic were affected by the test conditions. The Mann-Whitney test revealed that the differences were primarily found between conditions MA and IVE. This is an indicator that the participants experienced a higher level of negative emotion under the interactive condition. The emotions that showed increases under IVE condition were largely the ones that were intended as the emotional response of the horror-story, so detecting a difference indicates that the emotional involvement in the narrative was affected by the test condition. The statistically significant differences were detected mostly between conditions MA and IVE, which, along with the results from the IMX, indicates that condition IVE was a more effective experience.

The fact that DEQ and IMX results both indicated a small increase in immersive and emotional responses might be viewed as an indicator that the proposed interactions had an influence on the participants' overall experience of the story and the prototype. An influence to this can be caused by several factors:

- (i) An increased feeling of being present by doing something practical with your hands

- (ii) The freedom of head movement
- (iii) The spatial sound
- (iv) Combination of previous
- (v) Confounded, such as the state of mind of the participants before testing

One can eliminate, or try to restrict and contain, these confounding variables, by asking the participants to rate their current emotions before engaging in the test. For some of the participants, another confounder could be the usage of generic HRTFs, which might provide acoustic information too far from individual contribution of listener's body. Such discrepancy might lead to bad localization performances that resulted in a biased evaluation of the system like in the VR experience reported by Geronazzo et al. [49].

The results from the remote's logging data showed some different behaviors among the participants. All of them were adjusting the volume to positive and negative levels at the beginning minutes of the test, where some remained at an approximately fixed volume afterward, and others continue to use the remote continuously.

One of the primary problems with our test was the small sample size. Normally, a sample size of 28 for each group is needed to detect a large effect size in a between-subjects experimental design [50], but this experiment only had a sample size of 10. A larger sample size would be less prone to outliers and thereby reduce the risk of getting Type I errors.

The quality of the prototype was a critical aspect in providing a good representation of interaction, as a dysfunctional solution could severely affect the results of the test as opposed to the expectation. Another crucial variable of our study was the implementation of the narrative, as the chosen story was not originally designed as an interactive audio experience. The story, originally a written short story, has been adapted to fit the medium and desired duration of the experience. This means that the quality of the narrative could possibly have been decreased as the adaptation was not performed by a trained professional. Similarly, with the narrative, if the sound design is not optimal with the wanted experience, and immersiveness, the test results will have subjected to a negative effect as well. Finally, the heart rate monitor E4 Empatica provided some readings that were on unreasonable levels, as too high or too low, which had an influence on the analysis and interpretation of the data. These data points were removed from the data samples, so the overall amount of data is less than what was actually collected.

Another confounding variable was the implementation of the narrative, as the chosen story was not originally designed as an interactive audio experience. The story, originally a written short story, has been adapted to fit the medium and desired duration of the experience. This means that the quality of the narrative could possibly have been decreased as the adaptation was not performed by a trained professional.

Moreover, the heart rate monitor E4 Empatica provided some readings that were on unreasonable levels, as too high

or too low, which had an influence on the analysis and interpretation of the data. These data points were removed from the data samples, so the overall amount of data is less than what was actually collected.

In future development, the story selection could also be optimized and reviewed upon. The story can either be lengthened or shortened, depending on the goal in the evaluation. If one was to make another short 10 minute story, as done in this study, perhaps the story should be more intense and fast paced. If a longer 40 minute story was created, the story should be more intriguing in the long run and have several narrative atmospheres to keep the user actively listening.

9. Conclusion

The research on interactive storytelling revealed that both storytelling and interactivity can take several shapes. Within the story, the narrative can be linear or nonlinear and when it comes to interactivity, there are different dimensions to consider, like making sure the user has the ability to make choices. For this work, it was adopted a first-person linear narrative to make sure all the users were exposed to the same story while testing, in order to be able to use the data gathered. Our physical interface design revealed that users felt comfortable and natural in navigating a virtual world through auditory feedback with head-tracking. The quality of the experience was assessed by focusing on invoking a particular feeling in the user, as quantitative physiological measurements can be used for measuring this. Using a combination of heart rate monitoring and follow-up questionnaires was possible to determine the level of fear-induced emotion experienced, providing insight into the user's emotional involvement in the narrative. Our results suggested that dynamic binaural audio rendering allowed a greater level of immersion and more detectable fear-related emotions.

The possibility of an even further interactive story could also potentially be implemented. While interaction was incorporated in this study through head movement and a simple gesture control via hand-help controller, there might be more directions for increasing interactivity. In further studies, the user could be able to choose different paths in the storyline. This could allow the user to build the story themselves and therefore feel more involved in the process. This could be done by using the interactive remote as a selection tool, pointing in the story direction that they want to follow. This would also involve some form of crossroad areas of the story, where the narration would pause and the user would be instructed to choose between options. It is worthwhile to notice that costs and feasibility of new interactions will become more and more similar to those of computer games. The correct balance between passive and active listening should be carefully taken into consideration.

The integration of the proposed system in mobile devices for audio augmented reality (AAR) will open opportunities of new forms of spatial interactions and HRTF personalization [51]. The real deployment of such kind of audio story will require the development of story map in Unity and a definition of design guidelines for audio VE. On the other hand,

the technological platform for the required interactions should take advantage from available head-mounted displays trying to integrate audio-specific features into immersive VEs.

Data Availability

The data from the experimental sessions used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the internationalization grant of the 2016–2021 strategic program “Knowledge for the World” awarded by Aalborg University to Michele Geronazzo.

Supplementary Materials

In the supplementary material, we provide the shortened narrative of Stephen King's Strawberry Springs and the storyboard built upon it for our study. Moreover, a map of all created audio objects depicts the complexity of the virtual auditory scene designed following such a storyboard. (*Supplementary Materials*)

References

- [1] J. Richards, *Audiobooks Continues Double-Digit Growth—2017 Sales Survey*, Audio Publishers Association, Princeton Junction, NY, USA, 2017.
- [2] M. Furini, “Beyond passive audiobook: how digital audiobooks get interactive,” in *Proceedings of the 2007 4th IEEE Consumer Communications and Networking Conference*, pp. 971–975, Las Vegas, NV, USA, January 2007.
- [3] R. Nordahl and N. C. Nilsson, “The sound of being there: presence and interactive audio in immersive virtual reality,” *The Oxford Handbook of Interactive Audio*, Oxford University Press, Oxford, UK, 2014.
- [4] D. R. Begault, *3-D Sound for Virtual Reality and Multimedia*, Academic Press Professional, Inc., San Diego, CA, USA, 1994.
- [5] S. Serafin, M. Geronazzo, C. Erkut, N. C. Nilsson, and R. Nordahl, “Sonic interactions in virtual reality: state of the art, current challenges and future directions,” *IEEE Computer Graphics and Applications*, vol. 38, no. 2, pp. 31–43, 2018.
- [6] P. Coleman, A. Franck, J. Francombe et al., “An audio-visual system for object-based audio: from recording to listening,” *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 1919–1931, 2018.
- [7] C. Kim, R. Mason, and T. Brookes, “Head movements made by listeners in experimental and real-life listening activities,” *Journal of the Audio Engineering Society*, vol. 61, no. 6, pp. 425–438, 2013.
- [8] E. Degli Innocenti, M. Geronazzo, D. Vescovi et al., “Mobile virtual reality for musical genre learning in primary education,” *Computers & Education*, vol. 139, pp. 102–117, 2019.

- [9] I. M. Konstantakos, "The magical transformation contest in the ancient storytelling tradition," *Cuadernos de filología clásica: Estudios griegos e indoeuropeos*, vol. 26, pp. 207–234, 2016.
- [10] C. Crawford, *Chris Crawford on Interactive Storytelling*, New Riders, Carmel, IN, USA, 2005.
- [11] M. O. Riedl and R. M. Young, "From linear story generation to branching story graphs," *IEEE Computer Graphics and Applications*, vol. 26, no. 3, pp. 23–31, 2006.
- [12] J. Preece, H. Sharp, and Y. Rogers, *Interaction Design: Beyond Human-Computer Interaction*, John Wiley & Sons Inc., Chichester, UK, 2015.
- [13] S. M. Lwin, "Narrativity and creativity in oral storytelling: co-constructing a story with the audience," *Language and Literature*, vol. 26, no. 1, pp. 34–53, 2017.
- [14] M. Geronazzo, F. Avanzini, and F. Fontana, "Auditory navigation with a tubular acoustic model for interactive distance cues and personalized head-related transfer functions," *Journal on Multimodal User Interfaces*, vol. 10, no. 3, pp. 273–284, 2016.
- [15] M. Geronazzo, E. Peruch, F. Prandoni, and F. Avanzini, "Applying a single-notch metric to image-guided head-related transfer function selection for improved vertical localization," *Journal of the Audio Engineering Society*, vol. 67, no. 6, pp. 414–428, 2019.
- [16] T. R. Agus, S. J. Thorpe, and D. Pressnitzer, "Rapid formation of robust auditory memories: insights from noise," *Neuron*, vol. 66, no. 4, pp. 610–618, 2010.
- [17] D. Hammershøi and H. Møller, "Methods for binaural recording and reproduction," *Acta Acustica United with Acustica*, vol. 88, no. 3, pp. 303–311, 2002.
- [18] S. Paul, "Binaural recording technology: a historical review and possible future developments," *Acta Acustica United with Acustica*, vol. 95, no. 5, pp. 767–788, 2009.
- [19] J. Lewald, "Exceptional ability of blind humans to hear sound motion: implications for the emergence of auditory space," *Neuropsychologia*, vol. 51, no. 1, pp. 181–186, 2013.
- [20] M. Geronazzo, A. Bedin, L. Brayda, C. Campus, and F. Avanzini, "Interactive spatial sonification for non-visual exploration of virtual maps," *International Journal of Human-Computer Studies*, vol. 85, pp. 4–15, 2016.
- [21] F. Heller, A. Krämer, and J. Borchers, "Simplifying orientation measurement for mobile audio augmented reality applications," in *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems-CHI'14*, ACM, pp. 615–624, New York, NY, USA, May 2014.
- [22] W. Hess, "Head-tracking techniques for virtual acoustics applications," in *Proceedings of the Audio Engineering Society Convention 133*, San Francisco, CA, USA, October 2012.
- [23] J. Traer and J. H. McDermott, "Statistics of natural reverberation enable perceptual separation of sound and space," *Proceedings of the National Academy of Sciences*, vol. 113, no. 48, pp. E7856–E7865, 2016.
- [24] A. Andreasen, M. Geronazzo, N. C. Nilsson, J. Zovnercuka, K. Konovalov, and S. Serafin, "Auditory feedback for navigation with echoes in virtual environments: training procedure and orientation strategies," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 5, pp. 1876–1886, 2019.
- [25] G. Riva, F. Mantovani, C. S. Capideville et al., "Affective interactions using virtual reality: the link between presence and emotions," *CyberPsychology & Behavior*, vol. 10, no. 1, pp. 45–56, 2007.
- [26] A. Gorini and G. Riva, "Virtual reality in anxiety disorders: the past and the future," *Expert Review of Neurotherapeutics*, vol. 8, no. 2, pp. 215–233, 2008.
- [27] H. Sauzéon, P. Arvind Pala, F. Larrue et al., "The use of virtual reality for episodic memory assessment: effects of active navigation," *Experimental Psychology*, vol. 59, no. 2, pp. 99–108, 2012.
- [28] C. Huber, N. Rober, K. Hartmann, and M. Masuch, "Evolution of interactive audio books," in *Proceedings of the 2nd Conference on Interaction with Sound (Audio Mostly)*, pp. 166–167, Ilmenau, Germany, September 2007.
- [29] E. Marchetti and A. Valente, "Interactivity and multimodality in language learning: the untapped potential of audiobooks," *Universal Access in the Information Society*, vol. 17, no. 2, pp. 257–274, 2018.
- [30] S. Rubin and M. Agrawala, "Generating emotionally relevant musical scores for audio stories," in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology-UIST'14*, pp. 439–448, ACM, Honolulu, Hawaii, USA, October 2014.
- [31] D. Hilviu and A. Rapp, "Narrating the quantified self," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers-UbiComp'15*, pp. 1051–1056, ACM, Osaka, Japan, September 2015.
- [32] M. Grimshaw, "Sound and immersion in the first-person shooter," *International Journal of Intelligent Games & Simulation*, vol. 5, no. 1, 2008.
- [33] E. Gallo and N. Tsingos, "Efficient 3d audio processing with the GPU," in *Proceedings of the ACM Workshop on General Purpose Computing on Graphics Processor*, p. 1, ACM, Los Angeles, CA, USA, August 2004.
- [34] R. McGloin, K. Farrar, and M. Krcmar, "Video games, immersion, and cognitive aggression: does the controller matter?," *Media Psychology*, vol. 16, no. 1, pp. 65–87, 2013.
- [35] P. Skalski, R. Tamborini, A. Shelton, M. Buncher, and P. Lindmark, "Mapping the road to fun: natural video game controllers, presence, and game enjoyment, mapping the road to fun: natural video game controllers, presence, and game enjoyment," *New Media & Society*, vol. 13, no. 2, pp. 224–242, 2011.
- [36] D. Weibel and B. Wissmath, "Immersion in computer games: the role of spatial presence and flow," *International Journal of Computer Games Technology*, vol. 2011, Article ID 282345, 14 pages.
- [37] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience Harper Perennial Modern Classics*, HarperCollins, New York, NY, USA, 1st edition, 2008.
- [38] T. Björner, *Qualitative Methods for Consumer Research*, Hans Reitzel's Publishers, Kopenhagen, Denmark, 2015.
- [39] T. Garner, M. Grimshaw, and D. A. Nabi, "A preliminary experiment to assess the fear value of preselected sound parameters in a survival horror game," in *Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound, AM '10*, pp. 10:1–10:9, ACM, Piteå, Sweden, September 2010.
- [40] D. Spielberg Charles and C. Reheiser Eric, "Assessment of emotions: anxiety, anger, depression, and curiosity," *Applied Psychology: Health and Well-Being*, vol. 1, no. 3, pp. 271–302, 2009.
- [41] N. Curran, *The psychology of immersion and development of a quantitative measure of immersive response in games*, Ph.D. thesis, University College Cork, Cork, Ireland, 2013.

- [42] L. J. Norman and L. Thaler, "Human echolocation for target detection is more accurate with emissions containing higher spectral frequencies," *i-Perception*, vol. 9, no. 3, article 2041669518776984, 2018.
- [43] J. Panksepp, *Affective Neuroscience: The Foundations of Human and Animal Emotions*, Oxford University Press, Oxford, UK, 2004.
- [44] C. Harmon-Jones, B. Bastian, and E. Harmon-Jones, "The discrete emotions questionnaire: a new tool for measuring state self-reported emotions," *PLoS One*, vol. 11, no. 8, Article ID e0159915, 2016.
- [45] D. M. Anderson, *Mosby's Medical, Nursing, & Allied Health Dictionary*, Mosby, St. Louis, MO, USA, 2002.
- [46] R. Wang, G. Blackburn, M. Desai et al., "Accuracy of wrist-worn heart rate monitors," *JAMA Cardiology*, vol. 2, no. 1, pp. 104–106, 2017.
- [47] J. Pietilä, S. Mehrang, J. Tolonen et al., "Evaluation of the accuracy and reliability for photoplethysmography based heart rate and beat-to-beat detection during daily activities," in *Proceedings of the EMBEC & NBC 2017, IFMBE*, pp. 145–148, Springer, Singapore, 2017.
- [48] D. R. Begault, E. M. Wenzel, and M. R. Anderson, "Direct comparison of the impact of head tracking, reverberation, and individualized head-related transfer functions on the spatial perception of a virtual speech source," *Journal of the Audio Engineering Society*, vol. 49, no. 10, pp. 904–916, 2001.
- [49] M. Geronazzo, E. Sikström, J. Kleimola, F. Avanzini, A. De Götzen, and S. Serafin, "The impact of an accurate vertical localization with HRTFs on short explorations of immersive virtual reality scenarios," in *Proceedings of the 17th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 90–97, IEEE Computer Society, Munich, Germany, October 2018.
- [50] A. Field and G. Hole, *How to Design and Report Experiments*, Sage Pubns Ltd., Thousand Oaks, CA, USA, 2003.
- [51] M. Geronazzo, J. Fantin, G. Sorato, G. Baldovino, and F. Avanzini, "Acoustic selfies for extraction of external ear features in mobile audio augmented reality," in *Proceedings of the 22nd ACM Symposium on Virtual Reality Software and Technology (VRST 2016)*, pp. 23–26, ACM, Munich, Germany, November 2016.

Research Article

A Presence- and Performance-Driven Framework to Investigate Interactive Networked Music Learning Scenarios

Stefano Delle Monache ¹, **Luca Comanducci** ², **Michele Buccoli**,² **Massimiliano Zanoni**,² **Augusto Sarti**,² **Enrico Pietrocola**,³ **Filippo Berbenni**,³ and **Giovanni Cospito**³

¹Department of Architecture and Arts, IUAV University of Venice, Venice, Italy

²Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy

³Department New Technologies and Musical Languages, “G. Verdi” Music Conservatory, Milano, Italy

Correspondence should be addressed to Stefano Delle Monache; stefano.dellemonache@iuav.it

Received 7 February 2019; Revised 1 July 2019; Accepted 18 July 2019; Published 26 August 2019

Guest Editor: Michele Geronazzo

Copyright © 2019 Stefano Delle Monache et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cooperative music making in networked environments has been subject of extensive research, scientific and artistic. Networked music performance (NMP) is attracting renewed interest thanks to the growing availability of effective technology and tools for computer-based communications, especially in the area of distance and blended learning applications. We propose a conceptual framework for NMP research and design in the context of classical chamber music practice and learning: presence-related constructs and objective quality metrics are used to problematize and systematize the many factors affecting the experience of studying and practicing music in a networked environment. To this end, a preliminary NMP experiment on the effect of latency on chamber music duos experience and quality of the performance is introduced. The degree of involvement, perceived coherence, and immersion of the NMP environment are here combined with measures on the networked performance, including tempo trends and misalignments from the shared score. Early results on the impact of temporal factors on NMP musical interaction are outlined, and their methodological implications for the design of pedagogical applications are discussed.

1. Introduction

As distributed and ubiquitous interactive applications are increasingly populating our daily environments, we find ourselves constantly engaged in making sense of geographically displaced social practices and behaviors for the purpose of communicating, sharing, and cooperating. The network, in fact, is progressively evolving from a medium of communication, to a shared space virtually inhabited by bodily presences. Cooperative music making is a form of social practice characterized by peculiar temporal and spatial relationships, and in networked music performance (NMP), such relationships are unavoidably altered by the interposition of the network [1].

Computer systems for networked musical interaction have been categorized according to their temporal

(synchronous vs. asynchronous) and spatial (co-located vs. remote) dimensions of the performance [2], whereas a wide part of the scientific literature has focused on the technological and perceptual issues in real-time performance between musicians located in remote rooms, and requiring the highest degree of synchronicity, typically over teleconference-based communication media [3–5]. From a different angle, for years, research in telematic arts and music has been questioning the NMP communication model, stressing instead the role of the body in the network, and proposing an interpretive model of the emerging space, wherein questions of presence, plausible representations, sense of agency, and flow become crucial [6–8]. According to this view, telematic systems are conceived as proper instruments, whose apparent limitations are exploited instead in terms of creative affordances, for music purposes and interactions [9].

At the intersection between such disciplines, we witness a renewed interest in NMP technologies for music education [10]. NMP tools, in fact, are progressively being introduced in the market, and some solutions are emerging as viable standards in blended and distance learning, that is, a knowledge delivery combined with computer-mediated instruction, synchronous and face-to-face, or asynchronous [11]. The NMP approach to higher music education and pedagogy is the main locus of interest of this paper. Understanding how the use of mediating technologies affects the learning activity and experience becomes critical not only to inform the design and the development of learning environments and setups but also to adopt the appropriate pedagogical strategies [12–14].

The EU-funded project InterMUSIC (Interactive Environment for Music Learning and Practicing, 2017–2020) aims at bringing the solutions developed in NMP research within the context of distance education. The objective of the project is to distill a systematized knowledge in the form of best practices and guidelines for the design of remote environments for music interaction and higher education. Within the project activities, InterMUSIC is authoring three online pilot courses: music theory and composition, chamber music practice, and vocal training. The tools for the courses are developed according to two main paradigms. The former is based on MOOC approaches, namely, massive open online courses, wherein the learning experience is essentially individual and based on asynchronous communication with the master and limited collaboration with peers [15]. In the latter, students have access to NMP environments for attending master-student lessons or rehearse together. This paradigm clearly exhibits stricter requirements in terms of synchronicity and enables one-to-one (or few-to-few) synchronous communication. In this article, we focus on the NMP-based paradigm and we discuss the requirements related to the specific scenario of the chamber music practice. In particular, we approach the issue of synchronicity in NMP as a necessary condition for effective musical interaction, by focusing on how rhythm, melody, and expression affect the interaction and communication between performers. The existing literature on NMP pinpointed several factors affecting the objective quality of the music performance, namely, the unavoidable network latencies [16] and the effect of the resulting temporal separation of action and feedback [17], the timbral properties of the instruments [18], and the rhythmic complexity of the performed pieces [19]. These constitute the point of departure towards an interaction-centered and situated research approach to the case of learning and practicing chamber music in the NMP environment [20].

In the InterMUSIC project scenario, the goal is not to assess the NMP's objective quality, but rather to understand and frame those elements that can enable a comfortable remote rehearsal in teacher-student and student-student communication. Expert musicians are used to coping with the most adverse and diverse performing conditions, by relying on strong sensory-motor associations and by adapting their action planning in response to altered auditory feedback as much as possible [21, 22]. From this

viewpoint, much of the InterMUSIC research on NMP pedagogy is addressed to understand the design of learning environments in which the disruptive effects of the temporal and spatial alterations, due to the inherent physical distance and remoteness of performers, can be minimized or compensated through specific training and exercises.

In this article, we conceptualize the many forms of the networked music performance in a framework situated in the chamber music education scenario. Yet, the framework is abstracted enough to design and conduct our experimental activity and develop the pedagogical scenarios, and therefore it is the choice of appropriate technological solutions. In other words, we generalize the NMP actors and roles, at the same time narrowing the field of inquiry to the chamber music case, that is, we expect, for instance, findings not necessarily valid or prescriptive to other types of music practices (e.g., jazz improvisation). To this end, the emerging conceptual framework equally considers both the user experience, via presence constructs, and performance quality perspectives. In other words, we make use of metrics which describe the musicians' response to the NMP system and objective quality metrics of their performance as hypotheses generator for understanding and framing the design problem.

To operationalize our approach, we outline a pilot study, aimed at understanding how temporal factors (i.e., network latency) affect the interaction and communication of chamber music duos involved in remote collaboration for music making. The premise is that effective music making and communication rely on the availability of auditory and visual cues (i.e., sonic gestures) [23], which are inevitably constrained/limited in NMP and in telepresence environments in general. In other words, we explore how temporal alterations in audio-visual feedback affect the co-representation and coordination of ensemble musicians and seek for design strategies to compensate and facilitate a plausible music experience in the mediated environment. In the practice, we ask chamber music duos to perform a short exercise, under diverse conditions of network delay. The exercise is specifically conceived around musical structures which are functional to pinpointing a set of problems relative to time management, communication mechanisms, and mutual understanding between remote performers. The co-representation and coordination of duos are investigated by means of a presence questionnaire [24] and combined with objective metrics of the quality of the performance [19]. In this respect, we are exploring how and whether simulated network delay affects the subjective experience of being simultaneously present in the shared reality environment [25]. Sensory breadth and depth, degree of control, and anticipation of events, together with the overall interactivity of the environment, represent crucial elements in both presence and performance, being the first a prerequisite for the second [26].

The goal of this pilot study was to set the experimental playground, by exploring the NMP music learning scenario from a perspective (i.e., the presence experience), which is often recalled in the NMP literature, but that has never been systematically investigated [7]. In this respect, the 5 sessions collected do not make the underway study mature enough to

provide conclusions supported by statistical evidence. On the contrary, the information that we are gathering is instrumental to conceptualizing the relevant temporal (and spatial) factors affecting the NMP experience, from both the presence and performance perspectives. In other words, the exploratory framework, discussed in this paper, sets the reference for the experimentation with a more appropriate, extensive number of duos. Finally, the follow-up study will be addressed instead to the investigation of spatial representations, auditory and visual.

The paper is organized as follows: we first introduce the conceptual framework for NMP research in the classical chamber music scenario, in the context of presence studies and NMP music interaction; we then describe the experimental study, the design of the related questionnaire, and the selection of the quality metrics. Finally, we reflect on the methodological and design implications, by providing a narrative of the early results that we are collecting.

2. A Framework for InterMUSIC Research

In this section, we frame the relevant literature for the design of the chamber music practice and learning scenarios and experiments. We essentially look at that rich corpus of research focused on the temporal and spatial aspects, network constraints, and sound reproduction strategies, which have been considered the main factors affecting the sensory and control dimensions involved in the experience of playing together in a remote fashion [27].

In InterMUSIC, we approach the problem of designing an effective communication and interaction environment for networked music practice and teaching, by looking also at studies on presence [26, 28, 29] in the embodied cognition framework [24, 30]. The basic assumption is that the experience of music emerges in interaction, as complex network of predictive models of observable patterns and intentional states, which are acquired through knowledge and skills [31].

The literature overview is distilled in a conceptual framework of the NMP chamber music scenario, which is functional to devising and designing our experiments on the NMP performance as a whole and in its constitutive elements, i.e., actors, environment, and medium.

2.1. Temporal and Spatial Factors in NMP. The continuity of the time dimension in both remote music performance and tuition (i.e., teaching) is a necessary perceptual condition [32], yet the two activities are different in nature and call for a different time flexibility in system response. While in music tuition, conflicts and breakdowns are managed with the time flexibility of turn-taking in conversation [33], remote playing demands more extreme responsivity. The immediacy of responses between remote performers is essentially altered by the presence of end-to-end delays, dependent on the signal processing, throughout the whole chain, and network delay [4, 5].

The resulting latency level represents a factor that dramatically affects the NMP experience. It has been shown that

latency values in the range of 20–60 ms degrade the quality of the performance, by causing progressive slowdowns in the performance [19]. The effect of unnatural latencies has been explained in terms of incongruent temporal separation between action and auditory information, with respect to the expected “natural” time delays occurring in co-located performances [16, 17, 34]. Rhythmic clapping tasks have been used to assess the impact of latency (in the range of 3–78 ms) on the performance of two musicians playing in the network. It was found that best results are achieved when end-to-end delays approximate a temporal separation corresponding to a setting where the two subjects are in the same room. It was also observed that unnaturally low latency values (i.e., $t \leq 10$ ms) cause an acceleration in the musicians’ tempo, which can be explained in terms of sensory-motor synchronization by anticipation [35].

Indeed, these findings stress the importance of considering NMP as culturally situated practice in which bodily schemas are inherently part of the musical meaning and embodied means through which making sense of the networked space [7]. Both music playing and tuition rely on a significant repertoire of nonverbal forms of communication, and in the chamber music context, the types of body language and mutual cues embody the typical semicircular seating arrangement and the musical interpretation of the shared score. Typically, chamber musicians do not need to stare at each other, yet they support mutual understanding and negotiate the performance through glances and peripheral vision.

It has been argued that live video streams (i.e., full frontal views) of the musicians are essentially superfluous for the sake of the co-performance and that they respond to communication needs towards the listeners/observers, rather than between the performers [8]. While this observation is certainly embraceable, we hypothesize that properly designed spatial elements in remote interactive environments may facilitate the compensation of time-dependent misalignments in the performance and communication. In this respect, the design of the stage and the performance scene implies not only the physical displacement of various equipment in the rooms, but also the choice of appropriate solutions in both visual and sound feedback display [10, 33]. Various off-the-shelf strategies, such as life-size and near-life-size visual display to preserve a coherent distance perception in virtual proximity [36], projections and video loop techniques to support synchronous interaction [37], and the use of spatialized audio to increase the sense of presence [28, 38], have been proposed. The use of Ambisonic techniques has been explored to provide expressive and congruent environments in network music applications [39]. Studies in artificial reverberation rendering, in order to improve the realism, showed that virtual anechoic conditions lead to a higher imprecision than the reverberant conditions, while real-reverberation conditions lead to a slightly lower tempo, compared to the analog virtual reverberation conditions [40]. In general, the degree of spatial coherence needed is much dependent on the specific context and scope, whether the music interaction at hand accommodates loose coupling between performer and instrument, or demands close temporal coupling, with little

tolerance for interruption [41]. A minimum of spatial acuity and especially cross-modal congruence must be preserved, where the quality of the auditory content (i.e., timbral realism) represents a relevant cue through which the musicians negotiate their performance (e.g., agogics and instrumental blending) [28].

More recent research studies investigated the concept of Internet acoustics, which is the ambience resulting from the acoustical loop between internet endpoints, and the implementation of room-like Internet reverberators for collaborative music performances [42]. It should be noted, however, that the strict latency requirements imposed by the chamber music scenario make the adoption of any spatial audio rendering system very challenging. A possible solution could be to follow the route proposed in [43] and treat the rendering system as a distributed one, by taking advantage of a client-server architecture in order to meet the needed latency requirements and satisfy the need for spatial realism. Other works consider the avoidance of spatial realism and instead propose to take advantage of the spatial dissonance as a tool for music composition [44], which, however, is not suitable for the kind of music practice taken into account in this work. These audio-visual spatial aspects, however, are not considered in the present study and will be subject of separate investigation.

Finally, while we refer the reader to the extensive overview on NMP technologies discussed in [3, 5], here we briefly describe the three main alternatives of NMP software, as considered from the InterMUSIC perspective. JackTrip is the application developed by the SoundWIRE research group at CCRMA, to support bidirectional music performances [45]. It is based on uncompressed audio transmission through high-speed links such as Internet2. In the current version, it does not support video transmission.

The LOLA project was developed by the Conservatory of Music “G. Tartini” in Trieste, in collaboration with the Italian national computer network for universities and research (GARR) [46]. LOLA is based on low-latency audio/video acquisition hardware, optimized to transmit audio/video contents through a dedicated network connection. The main drawback is that the application is not open source and not serviceable for generic network connections.

On the other side, UltraGrid is an open-source project, whose application allows audio/video low latency transmission [47]. UltraGrid’s performance is not as effective as the LOLA environment, yet it represents a rather flexible solution for generic hardware use and networks connection and especially allows the development and integration of new functionalities. In the InterMUSIC project, we are currently using LOLA to conduct our experiments, although we are also considering the integration of UltraGrid in the next stages of the project.

2.2. A Presence Perspective on NMP. Presence is a multifaceted concept that has been interest of research since the early experiences in telepresence and virtual environments and more in general in computer-mediated environments applications [48]. The “feeling of being there” has been

emphasized as relevant factor in immersive mediated environments and assumed as construct that bridges the immersion quality of the mediating technology with the effectiveness of the mediated experience [49]. Ultimately, several definitions and models of presence have been proposed over the last two decades, with the aim of developing generalizable measures of mediated environment effectiveness and performance. A conclusive demonstration has not been provided yet [26], being the main problematic aspect the (lack of a) logical connection between presence, as an observable phenomenon, and task performance as a function of the user interface [50].

Presence frameworks provide, however, a consistent corpus of protocols, to evaluate the subjective experience in mediated environments, the immersive technology, and the quality of the interaction [26, 49]: validated self-reports in the form of postexperiment questionnaires focus on several presence components, including the individual characteristics of the user (e.g., attentional resources and attitudes), the coherence of the scenario (e.g., the realness and overall consistency of the mediated experience), and the immersion of the system (i.e., the set of sensorimotor and valid actions supported by the tracking system) [29, 51]. In addition, several presence-measuring techniques, behavioral and physiological (e.g., heart rate and skin conductance), have been proposed in order to capture the existence of the presence phenomenon in well-defined scenarios, in a rather objective fashion [25, 29]. Coherence and immersion can be controlled by design, to a certain extent, and developed to best reflect the user characteristics.

In the scope of InterMUSIC research, we make an instrumental use of the structural model of presence proposed by Schubert et al. [24], as resulted from the three-order factor analysis of 246 answers to a 75-item survey of questions taken from several established questionnaires on presence. Presence is defined as the cognitive feeling of being in a place [24]. This definition encompasses a concept of presence as embodied experience arising from the perceived self-location, the sense of agency, and the perceived action possibilities with respect to the specific scenario. The sense of presence is tied to action and active engagement in the (mediated) environment and results from the interpretation of the mental model derived. Schubert’s model of presence is composed of three main groups of components, which describe the subjective experiences of presence (spatial presence, attentional involvement, and realness), the evaluation of the immersive technology (quality of immersion, and dramatic involvement), and the evaluation of the interaction (interface awareness, ease of exploration, predictability, and interaction) ([24], p. 271). While the presence factors refer to the psychological experience, the other two groups pertain, respectively, to the presentation of the stimuli and the properties of the interaction. Although the psychological state and the immersion and interaction factors are separate constructs, Schubert claims that the relationship between the immersion and the sense of presence is not one-to-one; instead, it is the bodily and cognitive processes that mediate the impact of immersion on the experience of the psychological state of being in a place.

We agree with this view which fits the specific InterMUSIC scenario. This model of presence strongly resonates with the inherent sensorimotor nature of music communication and performance, where intentionality, corporeal articulations, interpretation, and perception of physical/sound energy represent the pillars of musical signification practices [52]: we see the NMP environment as a mediation technology layer which displaces the coupling of first-person descriptions (i.e., the mental representation of an intended musical gesture) and second-person descriptions (i.e., the corporeal articulation of the intended act), as an effect of the unavoidable alterations in the third-person description (i.e., the conditioning of physical/sound energy as returned by the networked auditory-visual capturing and display). In other words, we conceptualize the NMP environment as an instrument, on which chamber music instrumentalists project their direct involvement and mimetic skills.

On the other side, we also integrate the meta-analysis of presence models proposed in the literature, done by Skarbez et al. [29]. Presence is defined by the authors as the perceived realness of the mediated experience. The embodied, sensorimotor perspective proposed by Schubert (i.e., the cognitive feeling of being in place) is refocused on the balance of the three illusory experiences of (1) being in a place, in terms of action possibilities afforded by the system's immersion (the place illusion [51]); (2) the fidelity and plausible behaviors which make the apparent scenario to be coherent to the users/musicians (the plausibility illusion); and (3) the awareness of the copresence of another sentient being (e.g., the remote performer) and the extent to which a minimum degree of interaction gives rise to social behaviors (the social presence illusion).

The last point is worth a brief discussion, since playing music in ensemble is an essentially social and sharing activity, which entails complex sociocultural, coordination, and cooperation efforts and concerns [53, 54]. According to Skarbez, the social presence illusion refers to the "illusory (false) feeling of being together with and engaging with a real sentient being" ([29], p. 96:5): the term communicative immersion is proposed to trace those characteristics of the system that afford the transmission of communicative signals and ultimately the priming of the feeling of a warm, sensitive, and sociable medium (i.e., the *communicative salience*). Given the peculiarity of the chamber music NMP scenario, one may argue that musical connectedness and flow [55] represent the optimum illusory states of social engagement that skilled instrumentalists may expect to experience. In this respect, auditory immersion, especially in terms of spatial qualities and fidelity of sound, becomes crucial to maintain the sensorimotor loop and engage in plausible musical behaviors (e.g., instrumental blending) [28].

Taken together, we keep Schubert's presence components ([24], p. 279) and we group them according to the rationale of Skarbez's conceptualization ([29], p. 96:24). In addition, we remove the two factors of exploration and dramatic involvement, since they are not particularly meaningful in the frame of the chamber music scenario. Our main interest is not to actually define or model presence,

which is not the focus of InterMUSIC research; instead, we make use of presence studies as an opportunity to examine the chamber musicians' behaviors in the network, with the aim of providing a further perspective on NMPs.

The presence experience is composed then by three major groups of constructs, that is, components:

- (i) The spatial-constructive and attention facets of the experience of being there are, respectively, operationalized into *spatial presence* and *involvement* components
- (ii) The coherence of the scenario, that is, the reasonableness of the events primed to the user, is reflected into the *perceived realness* and *predictability* components
- (iii) The quality of the system's technology is distilled into the two components of the *interface awareness*, that is, distraction factors and the *quality of immersion*

Spatial presence (SP) refers to the emerging relation between the mediated environment as a space and the user's own body. The sense of "being in place" is tight to the role of the active body in constructing a spatial-functional model of the surrounding environment. The interpretation of the mental model emerges as cognitive representation of action possibilities in the actual environmental conditions, meshed with patterns of actions retrieved from memory. Spatial presence, that is, the place illusion (PI), implies a suspension of disbelief, about *how* the world is perceived, which can be ascribed to the immersion of the system, whereas the plausibility illusion (PsI) represents the orthogonal construct, about *what* is perceived [51]. In the realm of NMP, and particularly in the chamber music scenario, PI and PsI rather mirror the fidelity and consistency of the mediated environment, that is, the degree of real world emulation, in terms of coherent behaviors afforded. Indeed, the constraints of the basic, screen-based NMP setup may conceivably hinder the emergence of a spatial mental model of the networked environment.

Involvement (INV) or flow is a recurring concept in the presence literature and retains the attention side of the presence experience. The involvement component refers to the user's active engagement and focused attention and reflects "a psychological state experienced as a consequence of focusing one's energy and attention on a coherent set of stimuli or meaningfully related activities and events" [56]. Spatial presence and involvement are distinct and yet complementary psychological states, according to which individuals develop meaningful representations of a situation, by constructing the spatial mental model while actively suppressing conflicting sensory inputs. In the context of NMP chamber music practice, the involvement is reflected in the relative concentration on the real and the remote environments, in terms of focused allocation of attentional resources, concentration, and action awareness. In this respect, the degree of involvement is tightly connected to the coherence and consistency of the experience with the expectations from the real world.

Perceived realness (REA) encompasses reality judgments with respect to the meaningfulness and coherence of the scenario, as a function of the system's ability to provide stimuli which are internally consistent. From the perspective of NMP systems' development and design, it implies to consider appropriate and reasonable strategies to prime the musicians to expect certain types of behaviors and hence to develop adaptation and familiarity. Perceived realness and predictability provide a clue of the overall consistency and meaningfulness of the remote environment and represent a subjective evaluation of the interaction.

Predictability (PRED) refers to the possibility to anticipate what will happen next, in terms of activation of motor representations as a consequence of perceiving while playing. These include not only self-generated actions, but also the actions (and effects—which and when) produced by others [21]. The predictability in the chamber music performance is supported by the musical interpretation of the shared score. Similarly to the realness component, the predictability of the NMP system means that the environment should be consistent and undeviating in its behavior, in order to facilitate the musicians' adaptation. Sensory breadth, that is, the amount of sensory channels simultaneously presented, is expected to affect predictability, and hence the coherence of the scenario. Sensory breadth can be modulated in depth, that is, resolution. Higher resolution in the sensory modality essential to the task is expected to lead to more presence.

Interface awareness and quality (IA) take in account distraction factors, that is, the obtrusion of control and display devices in terms of interference in the task performance. This factor gives a different viewpoint on the selective attention ability and focused concentration of the user and in general provides a clue of the mastery of the interface in the specific activity at hand. In the NMP scenario, the interface quality refers to the spatial staging, and its acceptance may increase through practice and familiarity. Eventually, a high quality interface may have an impact on the performers' involvement.

Quality of immersion (QI) is a component related to the presentation of the stimuli and can be defined as the set of valid actions supported by the mediating environment [29]. It has been shown that high levels of immersion do not necessarily lead to a higher level of presence experience and that spatial immersive features can be more effective than higher quality sensory contents in leading the users to construct a spatial mental model of the environment (i.e., the place illusion) [49]. The immersion emerges as a quality of the system's technology, reified in the vividness, that is, the sensory breadth and resolution and the interactivity of the system, including the update rate and association of controls and displays, and the range of the interactive attributes available for active search and manipulation (e.g., the binaural sound rendering as a function of the performer's head motion).

It is evident that all these components are intertwined, and they mutually affect each other. The design and development of remote interaction environments may favour one or a combination of these constructs, depending on the

characteristic of the application. NMP environments are constrained by a music making task which is very specific and demanding, especially in the context of chamber music practice. Put in design terms, this represents an advantage, potentially providing clearer user requirements. Much of expected outcomes of the current research are aimed at collecting and organizing the instrumentalists' expectations. For example, if the goal of the music practice at hand is to actually rehearse a performance in the network, the induction of a state of involvement or flow might be preferable. Training applications may rather focus on provoking a plausible effect of social presence. Time continuity and connectedness become crucial. On the other hand, when the goal is to transfer the training to the real world, the coherence and fidelity of the scenario are likely most important.

The objective of this research track of the InterMUSIC project is to understand and conceptualize first the most relevant factors affecting the experience of studying and rehearsing music in a networked environment [57]. The experimental research is aimed at framing the design problem of an online learning environment for master students of chamber music. The overall goal is to define design guidelines for the staging of remote rooms in higher music education institutions, including the NMP system's technology.

2.3. The Framework Conceptualized. As shown in Figure 1, a *performance* occurs when two or more *subjects* musically interact together through a *medium*, in an *environment*. The performance is the entity at highest hierarchical level and may assume two main configurations, namely, a *performed music composition* or a *taught lesson*. The music performance can be of two main types, a *rehearsal* or a *concert*, and both involve the interaction of at least two peers, that is, the subjects are both musicians. In the lesson configuration, the performance involves subjects with different roles, a teacher and at least one student. When subjects interact in the same room, we have a *local performance*. However, in the InterMUSIC scenario, we consider either the case of geographically displaced subjects (*networked performance*), or the case of a *mixed performance*, when two or more subjects are accommodated in the remote rooms. In this respect, the spatial property of the performance is a function of the *medium*.

When subjects interact by means of a *physical* medium, that is, simple air propagation, the performance is local. In the case of networked performances, the medium is a *networked medium* and includes the Internet connection and the NMP software/hardware equipment to connect the subjects. Mixed performances involve both physical and networked media.

The environment is the space hosting the subjects, with its own specific timbral and spatial properties, i.e., the acoustics, the staging, and the subject(s) displacement in the room. In the case of networked and mixed performance, environments with different characteristics are potentially involved. Given a subject, we define the environment where

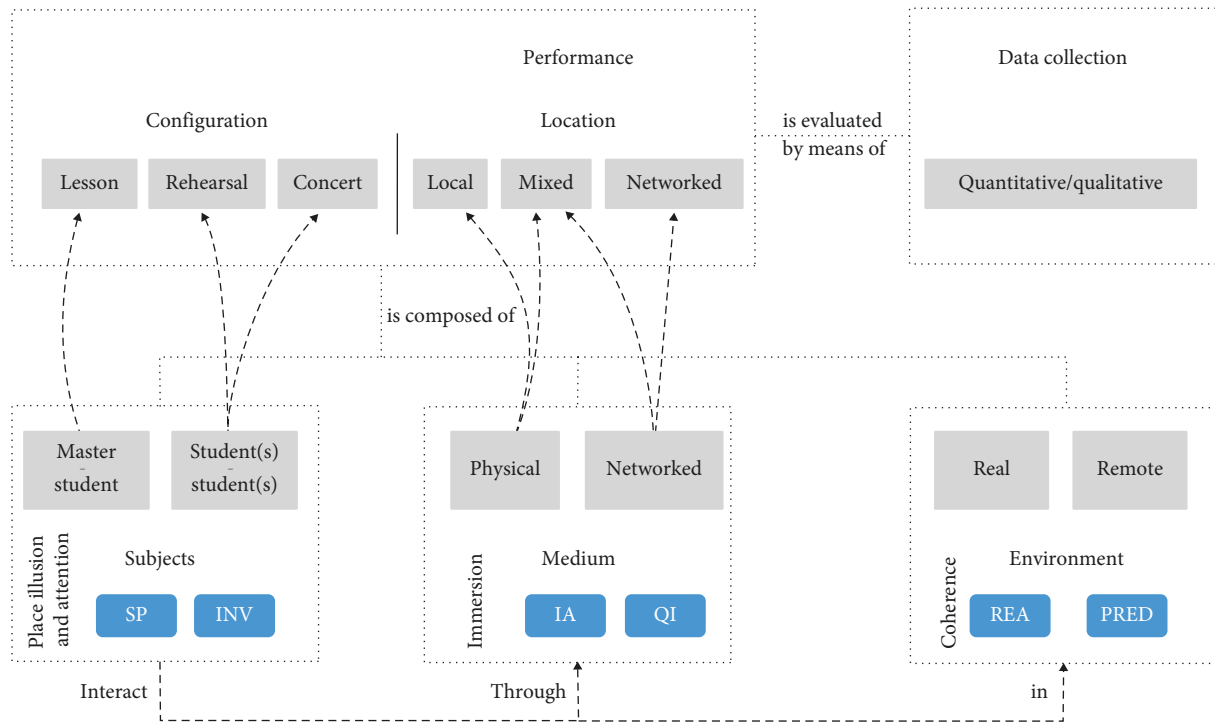


FIGURE 1: The conceptual framework for NMP research in the chamber music practice and learning scenario.

she is playing as the *real environment* and the representation of the environment relative to the geographically displaced co-performer as the *virtual or remote environment*. The performance unfolds in the situatedness of the networked space emerging from the real and the virtual environments. As an example, Figure 2 shows the typical basic NMP staging wherein the real environment of each musician is returned to the other co-performer in the form of a remote audio-video representation. The perceived coherence of the NMP scenario results from the overall congruity of the two environments.

Data collection is essential to the analysis of the performance as a whole and in its constitutive facets. Diverse approaches may come into play to capture the nuances of the music making experience in the networked environment, from the acquisition of multimodal signals to the collection of video-ethnographic observations and self-reports. The first distinction concerns the performance configuration, whether the networked space is meant to host a concert or a lesson. These two types represent the edges of continuum, wherein music interaction (i.e., playing) and gestural and verbal communication clearly have a different relevance and purposes. A performance can be described by date and time, location(s), type, metadata, MIDI or MusicXML symbolic representation, and the score (including its musicological analysis).

The concert type relies entirely on forms of communication, coordination, and leadership, based on musical interpretations previously negotiated and equally shared according to the score indications [58]. These kinds of performance represent a proper task, which is essentially addressed to the external world and experienced from a

second-person perspective, that is, a listener/observer. The concert is the configuration commonly investigated in NMP research and technological development [4, 5, 59]. Multimodal signal acquisition includes the audio recordings from both the remote environments, to compute several measures on the quality of the performance [19], gaze tracing and eye tracking to annotate the visual search of the musicians [60], and the biometric response of the performers for distress estimation [61].

In the lesson configuration, the teacher-student musical interaction rather follows the turn-taking mechanisms of conversations. Typically, musical coordination in the local lesson is maintained through verbalizations, visuospatial gestures, bodily postures, and especially joint annotations of the shared score [33]. However, when the medium is networked, the intrinsic disruptions may affect the management of verbal and musical interruptions and overlaps, despite the higher time flexibility of networked conversational turns [17]. Networked and environmental audio-video recordings of the teaching session can be annotated and parsed in salient conversational turns in order to conduct ethnographic and conversational analyses [62]. The rehearsal represents an intermediate situation, between the concert and the lesson. Synchronous and instantaneous music interaction intertwines with conversational turn-taking mechanisms, for the purpose of practicing or preparing a concert [63].

Self-reports and questionnaires are used to collect information on the quality of the performance, as perceived by the subjects, and in general on their musical experience in the networked space. We make use of a presence questionnaire, whose items reflects constructs that can be

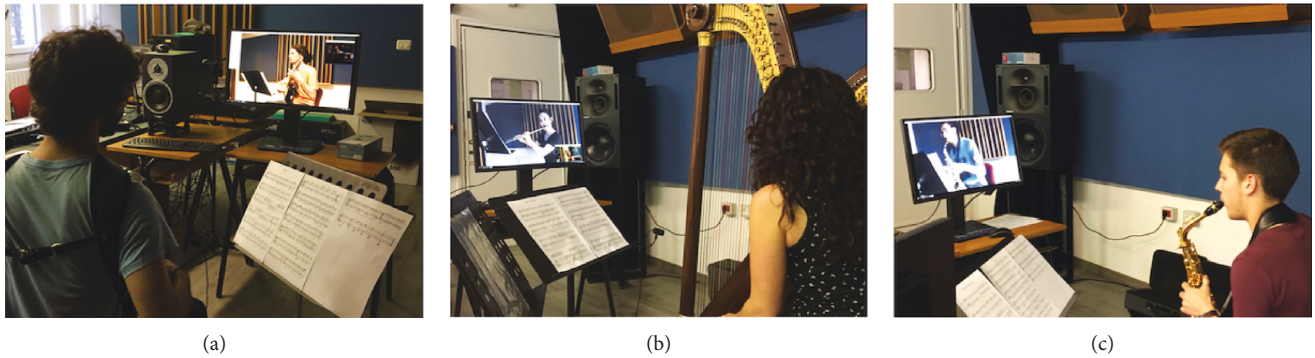


FIGURE 2: Instrumentalists in Room 1, with the frontal view (on screen) of the co-performer displaced in Room 2.

selectively ascribed to the musician herself (spatial presence (SP) and involvement (INV)), the medium (interface awareness (IA) and quality of immersion (QI)), and the environment (realness (REA) and predictability (PRED)).

Subjects are identified by name, age, experience, and musical background. In general, the user types considered in the InterMUSIC scenario are musicians with at least 5 years of academic music practice. The instrument is a relevant feature of the subject, and content-based analyses are used to integrate the user's profile [19]. Previous experiences with virtual environments and motivation are also relevant individual information to account an attitude to adaptability and selective attention.

The environment data include the spatial staging and acoustic properties of both real and remote places. Depending on the types of performance (i.e., concert, rehearsal, and lesson), the environment may exhibit different requirements [10, 62]. In this respect, properties of this entity are the audio/video capture and rendering devices configurations and their degree of experienced coherence with respect to the specific type of performance and scenario. By configuration, we mean the effects of design choices in the electroacoustic chain and signal processing, for example, the types of microphones according to their cost benefit [64], the moulding of a shared space based on Internet acoustics [42, 65], or acoustic scene analysis [66] and spatial rendering techniques over arrays of speakers [67] or headphones [68]. Design choices in visuo-spatial representations, i.e., video capturing and rendering, are mostly constrained by the requirements of the NMP system architecture employed [46].

Data relative to the medium essentially concern the system architecture, hardware and software, and network characteristics, that is, bandwidth and latency. From the subjects' viewpoint, the physical and the networked media affect in a different way the temporal separation between the remote performers. In the concert and rehearsal types, dynamic alterations of the temporal separation, caused by the network latency, result in disruptions of the mutual coordination of the ensemble. The medium features in the actual NMP environment qualify the immersion of the system's technology.

In the following Section, we introduce the pilot experiment [69], which represents an ongoing tool for reflection and conceptualization of the NMP chamber music practice and learning scenario.

3. Experimental Study

The objective of the study is to conceptualize the disruptive effects of network delay and interruption on time management, communication mechanisms, and mutual understanding between remote performers and frame them in the subjective experience of playing together in the networked space. In other words, a mixture of quantitative evaluation, based on objective performance quality metrics, and qualitative assessment and observations are aimed at putting in the foreground the users' needs and system requirements. The outcomes are expected to provide relevant design implications in the staging of classrooms dedicated to remote music practice and teaching. The current study focuses on the temporal dimension of the networked experience, while keeping constant the spatial dimension, i.e., strategies in auditory and visual spatial representation, which will be subject of forthcoming inquiry.

3.1. Method. The pilot experiment took place at the Conservatory of Music "G. Verdi" of Milano, in two dedicated rooms, equipped with direct network connection and all the necessary facilities. We asked couples of musicians recruited from the conservatory to perform a short exercise, under diverse conditions of network delay. A qualitative assessment through questionnaires on the sense of presence and the perceived quality of the performance [24] was combined with quality metrics of the objective performance [19]. The scores of the stimuli and the presence questionnaire are publicly available.

Ten volunteers (five duos, five males, five females, age ranging from 14 to 29, average age = 21.9 years, SD = 4.7) were recruited from the class of chamber music practice. They were all musicians with at least five years of academic musical practice. Each duo had already a familiarity of minimum two weeks of rehearsal. Table 1 reports the instruments and the performers' location, in rooms 1 or 2 (see Figure 3).

3.1.1. Presence Questionnaire. Self-reports in the form of postexperience questionnaires are common measurement means used in presence studies. Several questionnaire models have been proposed, which respond to the different models of presence, variously addressing specific constructs

TABLE 1: Instruments and their location in Room 1 or Room 2.

Room	Couple A	Couple B	Couple C	Couple D	Couple E
1	Mandolin	Accordion	Percussion	Harp	Alto sax
2	Mandolin	Guitar	Percussion	Flute	Alto sax

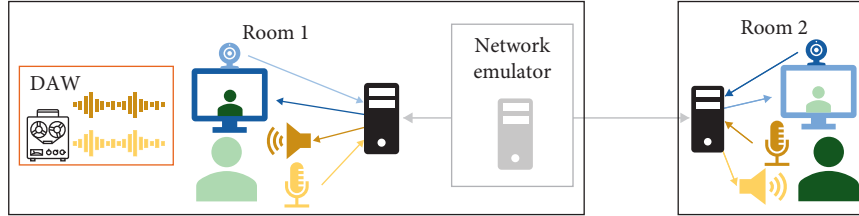


FIGURE 3: Setup of the pilot experiment, where two musicians perform together through a network emulator.

and situations in virtual reality (see [29] for a comprehensive review). One advantage of using published presence questionnaires is that they are the result of extensive validation, which makes them sensitive and reliable, yet the main drawback is that they are intrusive, not concurrent with the experience under inquiry, and prone to subjective (mis)interpretations of potentially difficult constructs. Furthermore, a clarification must be given: presence represents a complex phenomenon whose observable and measurable existence is still under debate, that is, the availability of questionnaires and related data does not demonstrate *per se* an effective measure of presence. In other words, attention must be paid in relying on questionnaires only to draw conclusions on the actual manifestation of the presence experience when interacting in a virtual environment [50]. In the InterMUSIC scenario, we use the information extracted from content-based analyses of the performances to derive observable patterns in music interaction and communication, and we approach the presence perspective to generate hypotheses on the relevant components that may affect the experience of two or more instrumentalists engaged in attending a master class, or rehearsing.

The questionnaire used in the study was constructed by merging three reference questionnaires on presence and selecting the most appropriate items with respect to the specific activity of performing music in the networked space (e.g., questions pertaining to navigation, objects motion, and manipulation in the virtual reality are not applicable to the case of NMP). In particular, we referred to the Witmer-Singer Presence Questionnaire [56, 70], and its revision by the UQO Cyberpsychology Lab [71], and especially the IPQ Igroup Presence Questionnaire by Schubert et al. [24].

Items were partially rephrased and adapted to the musical context and language. The resulting close-ended, 7-point Likert scale questionnaire was edited in Italian. The questionnaire is split in two main parts: a general post-experiment 27-item questionnaire organized in five sections and a postrepetition questionnaire with five questions extracted from the general one. We report the postexperiment questionnaire in Table 2 (with the median, mean, and standard deviation of the answers). The questionnaire is

devised around the three main constructs of presence, with an additional group of items focused on the performance:

- (i) Involvement and place illusion (Section 1): questions encompass the focused concentration and attention allocation to the music performance (Q1.2 and Q1.4–7) and the illusory feeling of connectedness with the remote environment (Q1.1 and Q1.3). It is hypothesized that in the NMP scenario, the attentional resources are more salient than the feeling of place illusion. For instance, while it can be argued that the sense of “being there” (i.e., the SP item, Q1.1) is of less utility in the screen-based NMP interaction, we retained this item exactly to question the suspension of disbelief about the networked space marked by the frontal views of the performers.
- (ii) Coherence (Section 2): this section reflects the perceived coherence of the scenario and includes the subscales of predictability (Q2.1–2, Q2.4–5, and Q2.7–8) and realness (Q2.3 and Q2.6).
- (iii) Immersion (Sections 3 and 4): the immersion quality of the system’s technology is retained in Sections 3 and 4, which, respectively, includes questions relative to distraction factors (interface awareness, Q3.1–5) and the vividness and interactivity of the NMP environment (Q4.1–3). The last set of questions is focused on the effect of the visual and auditory representations as a whole (i.e., screen-based frontal view and monoaural reproduction of the close take of the instrument) on the availability of various information, such as eye contact, foot tapping, breath attack, and instrumental blending, which in turn affect the musicians’ involvement.
- (iv) Quality of the performance (Section 5): this section is dedicated to the subjective assessment of the quality of one’s own performance (Q5.1–4).

Five items with a peculiar emphasis on the experience of delay have been extracted from the general questionnaire and included in the postrepetition questionnaire (items are emphasized in *italics* in Table 2).

TABLE 2: Statistics of the answers to the postexperiment questionnaire.

Question	Median	Mean	Std
<i>(1) Involvement and connectedness</i>			
(1.1) In the remote environment, I had a sense of "being there."	3.5	4.17	1.77
(1.2) The sense of playing in the remote environment was compelling.	5.0	4.50	1.50
(1.3) I had a sense of playing in the remote environment, rather than performing something from outside.	4.0	3.80	1.17
(1.4) How aware were you of the real world surroundings around you during the performance?	4.5	4.00	1.91
(1.5) How completely were you able to actively survey the musical environment using vision?	4.0	3.83	1.57
(1.6) How completely were you able to actively survey the musical environment using audition?	5.0	4.67	0.94
(1.7) The delay affected the sense of involvement.	4.5	4.00	1.63
<i>(2) Coherence</i>			
(2.1) The musical interaction in the remote environment seemed natural.	5.0	5.00	0.82
(2.2) The environment was responsive to actions that I performed.	4.0	4.00	1.83
(2.3) How much did your musical experience in the remote environment seem consistent with your real world experiences?	4.0	4.40	1.36
(2.4) I was able to anticipate the musical outcome in response to my performance in the remote environment.	5.0	5.00	1.00
(2.5) The environment was responsive to actions performed by my partner.	4.5	4.25	1.48
(2.6) How realistic did the remote environment seem to you?	4.5	4.33	0.75
(2.7) I was able to anticipate the musical outcome in response to the performance by my partner in the remote environment	4.0	3.83	1.21
(2.8) It was difficult to cope with the distance performance.	4.0	3.83	1.57
<i>(3) Interface awareness and quality</i>			
(3.1) How well could you concentrate on the music performance rather than on the mechanisms required to perform?	5.5	5.00	1.53
(3.2) How aware were you of the display and control devices/mechanism?	4.0	4.00	1.10
(3.3) How much did the visual display quality interfere or distract from performing?	3.0	3.17	1.21
(3.4) How much did the auditory display quality interfere or distract from performing?	6.0	5.17	1.46
(3.5) How much delay did you experience between your actions and expected outcomes?	4.5	4.33	1.60
<i>(4) Quality of the immersion</i>			
(4.1) The visual representation made me feel involved in the remote environment.	3.0	3.33	0.94
(4.2) The auditory representation made me feel involved in the remote environment.	4.0	4.00	1.73
(4.3) I felt involved in the remote environment experience.	5.0	4.60	0.49
<i>(5) Quality of the music performance</i>			
(5.1) How quickly did you adjust to the experience of playing in the remote environment?	4.5	4.17	1.67
(5.2) It was easy to cope with the delay to adjust the quality of the performance.	3.5	3.33	1.25
(5.3) How proficient in remote music playing did you feel at the end of the experience?	5.0	4.67	1.89
(5.4) The delay affected the quality of my performance.	5.0	4.33	1.80

Postrepetition items are highlighted in italics.

3.1.2. Objective Quality Metrics. A content-based analysis was performed on the audio recordings of the duos' performances, in order to derive an objective description of the quality. In the literature, researchers have proposed different metrics, related to the rhythmic trend of the performances [5]. Figure 4 provides an example of the annotation procedure that leads to the computation of these metrics. Given the stimulus represented by a score (Figure 4(a)), the first step is to annotate, in the recordings, the instants when an onset occurs on the beat, as t_1, t_2, \dots, t_N (Figure 4(b)). In order to address the issues of beats occurring without an onset, e.g., because of a four-quarter note, we also annotated the amount of beats occurring between the n -th instant and the following (t_n and t_{n+1}) as v_n . In Figure 4(b), for example, the onset occurring at t_{n+3} is followed by a two-quarter pause, hence the onset at t_{n+4} occurs after three beats and $v_{n+3} = 3$. We convert the set of annotations $(t_1, v_1), \dots, (t_N, v_N)$ to the tempo samples (in BPM) $\bar{\delta}(n) = (60 \cdot v_n) / (t_{n+1} - t_n)$.

Following the previous example, we show the final BPM annotation in Figure 4(c); we apply a 5-second moving average filter to $\bar{\delta}(n)$ to compensate the variance due to the musical agogics and the resulting imprecision of the manual annotation.

From the set of t_n , v_n , and $\bar{\delta}(n)$, we can compute several metrics [5] related to the tempo slope or the asymmetry between the two performers.

The *tempo* slope κ provides a compact descriptor of the tempo trend as the slope of its linear approximation. In particular $\kappa = 0$ when the tempo remains steady for the whole performance, and it assumes positive or negative values in case of acceleration or deceleration, respectively.

The *asymmetry* α provides a metrics of misalignment between the two performers, and it is strictly related to the beat and the score. Let us define $t_n^{(A)}$ and $t_n^{(B)}$ as the beat instants, corresponding to the execution of the two performers A and B playing during the same performance. Given the score of the performance, we can define a set of pairs $(n^{(A)}, n^{(B)})$ related to the beats common to both performances. This process is shown in Figure 5, where two example measures are shown, with the indications of the quarter onsets corresponding to $n^{(A)}$ and $n^{(B)}$. Not all quarter onsets are comparable; specifically, in the case shown in Figure 5, it is possible to compute the misalignment only between the annotated time instants $t_n^{(A)}$ and $t_n^{(B)}$ corresponding to the notes indexes $\{(n^{(A)}, n^{(B)})\} = \{(39.2^{(A)}, 39.2^{(B)}), (40.2^{(A)}, 40.2^{(B)})\}$. Now, we can define the intersubject time difference (ISD) between A and B as the time distance between the two instants:

$$t_n^{(AB)} = t_n^{(A)} - t_n^{(B)}. \quad (1)$$

If $t_n^{(AB)} < 0$, it means that in that particular beat, the performer A has anticipated performer B and vice versa, while $t_n^{(AB)} = 0$ indicates the performers played in the same instant. In order to obtain the asymmetry, we average the ISDs through parts of the performance, or through the whole performance, to have a more global descriptors of the interaction between the two performers. Let us define for our convenience the set of common beats $\mathcal{N} = \{(n^{(A)}, n^{(B)})$

$\},$ of size $|\mathcal{N}|$; then, we can write the asymmetry as

$$\alpha^{(AB)} = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} t_n^{(AB)}. \quad (2)$$

It must be noted that the value of the ISDs, and hence of the asymmetry, depends on the point of the recording. In our case, the recording was performed in room corresponding to the subject A (i.e., Room 1, in Figure 3). We can infer the ISDs of subject B using the information on the two-way latency λ , as

$$\begin{aligned} t_n^{(BA)} &= t_n^{(B)} - t_n^{(A)} = \left(t_n^{(B)} - \frac{\lambda}{2} \right) - \left(t_n^{(A)} + \frac{\lambda}{2} \right) \\ &= t_n^{(B)} - t_n^{(A)} - \lambda = -t_n^{(AB)} - \lambda. \end{aligned} \quad (3)$$

Note that this may lead to some contradictory behaviors. Suppose, for example, that $t_n^{(AB)} = -25$ ms (where we neglect the n for the sake of clarity), which means that performer A is anticipating performer B by 25 ms. However, if the two-way latency is $\lambda = 50$ ms, this means that in the room where B is performing, we measure $\alpha^{(BA)} = -\alpha^{(AB)} - \lambda = -25$ ms, hence performer B is also anticipating performer A .

The analysis of asymmetry, therefore, must be conducted analyzing both sides of the medium in order to draw meaningful considerations. The asymmetry $\alpha^{(BA)}$ corresponding to the point of view of the room where performer B is playing, can be computed as

$$\alpha^{(BA)} = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} t_n^{(BA)} = -\lambda - \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} t_n^{(AB)}, \quad (4)$$

where we use equation (3) for the second part of the formula.

The tempo slope κ and the asymmetry α enable us to consider two different aspects of the interaction between the musicians during the performance with an objective formulation.

3.1.3. Apparatus. The research activity took place in two dedicated rooms with direct connection at the Conservatory of Music of Milano. The rooms are two acoustically treated studios and are located in two different floors of the building with no direct sound interference. The experimental setup is shown in Figure 3.

The performance in each room was captured by means of one Audio Technica ATM350 cardioid condenser clip-on microphone applied to the instrument (monoaural acquisition) and a low latency Ximea MQ13CG-E2 USB 3.1 Gen 1 camera (with a Tamron TA-M118FM08 lens) placed in front of the performer and rendered by means of one Dynaudio BM5 mk3 7 "studio monitor loudspeaker (monoaural rendering) and a 27" 144 hz Asus ROG video monitor in the same frontal position of the camera. This basic staging reflects the current usage in remote music practice and tuition [46, 62]. Figure 2 shows the staging in Room 1. We consider this spatial arrangement as baseline for further investigations, assuming that it may represents the worst case scenario.

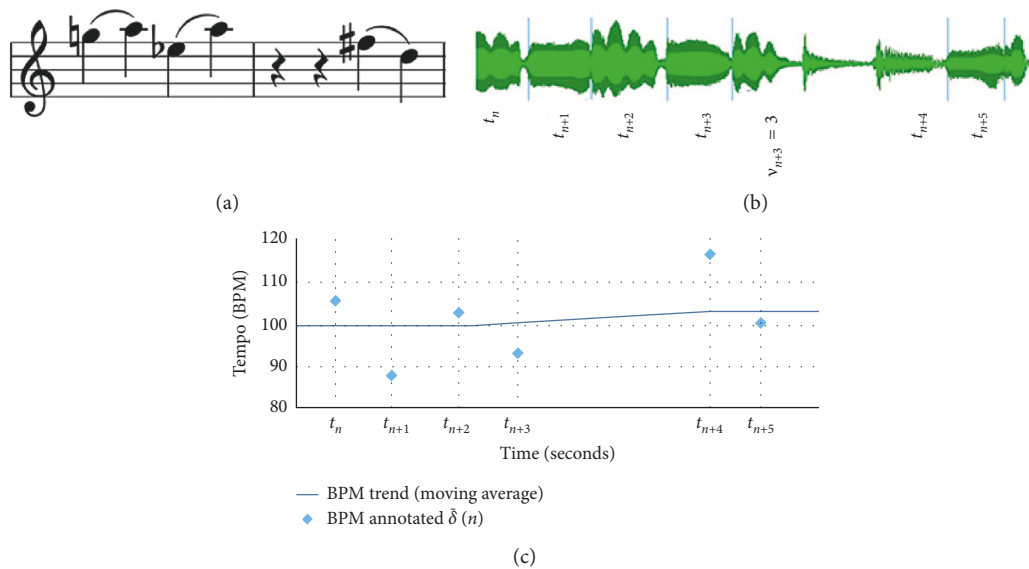


FIGURE 4: Representation of the annotation procedure for computing objective content-based metrics. (a) The stimulus. (b) The signal annotated with t_n and v_n . (c) The computed BPM trend $\bar{\delta}(n)$.

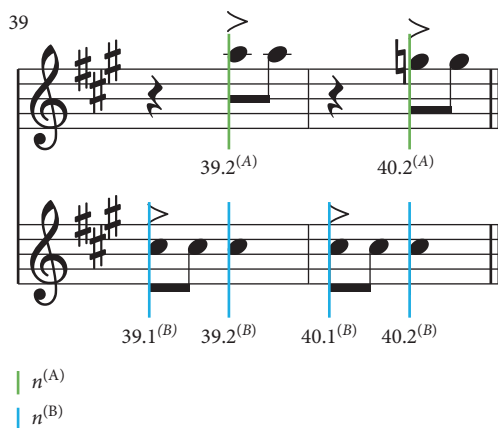


FIGURE 5: Example stimulus, with notes corresponding to quarter onsets suitable for the computation of the ISD.

The hardware equipment was connected to two computers, namely, two high-end Intel/Nvidia powered workstations with i7 esa/octa core processors, using PCIe audio cards, running Windows 10 OS, according to both LOLA and UltraGrid hardware and software requirements. The computers communicated together through a Gigabit Cat6 ethernet connection to a common server. The server, equipped with two Gigabit ethernet interfaces, acted as a Network Emulator to add a fixed delay to both audio and video streams. The effect of jitter, i.e., stochastic variation of the delay, will be considered in future experiments [19]. The server was placed in Room 1 to be easily accessible during the tests and to ease troubleshooting in case of network issues.

The audio output and audio input of the performer in Room 1 were redirected to a Digital Audio Workstation to record the performance (from the perspective of Room 1). These recordings were used to compute content-based metrics of quality, as described in the previous subsection.

Even when the latency was set to zero on the network emulator, a certain amount of it was still present, namely, the processing time, caused by the processing chain composed of analog-digital conversion, acquisition, preparation of packets, network stack, etc. In our experiment, we estimated the processing time by short-circuiting the output to the input in the equipment in Room 2, and we generated an impulse from Room 1 and recorded the delayed output, when the network delay was set to 0. We estimated the two-way processing as 28 ms, which can be seen (acoustically) as two musicians playing four meters apart.

3.1.4. Stimuli. The stimuli proposed to the musicians consisted of a score composed by one of the authors and were designed to take into account diverse basic structures of musical interaction in classical chamber music, with respect to time management and communication strategies. We have considered the chamber music duo as the basic instrumental group to approach different kinds of musical interaction. The rationale of the stimuli (i.e., the scores proposed to the duos) concerns simple, yet constraining aspects of synchronicity in musical time, as established in western music tradition, that is, the tight link between the musical dimensions of rhythm, melody, and expression. The objective was to direct the performers towards a complete musical interaction, leaving out any form of purely technical or quantitative test.

In this respect, we looked at Bartók's *Mikrokosmos* piano pieces [72], which represent a valuable methodological compendium of exercises in meaningful *rhythm-melody-expression* relationships: the didactic and technical purpose is immediately connected to the musical sense. In Table 3, we pinpoint eight types of musical structures which combine diverse expressive relationships of rhythm, melody, and expression (this expression includes musical markings

TABLE 3: Types of musical structures in rhythm-melody-expression relationship and examples of their combination present in the exercises in the score (leftmost column).

Ex.	Rhythm	Melody	Expression (dynamics, articulation, agogics)
3a	Homorhythm	Octave or unison	Static
8a	Homorhythm	Opposite direction	Static
5	Heterorhythm	Opposite direction	Alternation
7	Heterorhythm	Homodirection	Climax
6b	Phasing	Octave or unison	Static
4b	Slicing	Slicing	Dynamic
2b	Imitation	Imitation	Imitation
3b	Ostinato	Pedal tones	Static

regarding dynamics, articulation, and agogics). The leftmost column reports a few exemplary exercises as referenced in the score, to ease the reader’s understanding. From the perspective of rhythm, a musical structure can be *homorhythmic* or *eterorhythmic*, whether the articulation for each part is, respectively, the same and coincident or different. A *phasing* articulation occurs when the parts have the same rhythm, but their alignment is characterized by a short time delay. A *slicing* articulation refers to a musical phrase in which the rhythm as a whole is alternatively split between the parts. *Imitation* and *ostinato* refer to common repetition strategies in musical practice. The melodic structures essentially reflect the pitch directionality as articulated in the parts. Finally, the expression articulations between the parts can be *static* when there are no variations of expression markings, *alternated*, or arranged in a *climax*.

Figure 6 shows an example of homorhythmic, unison melody, with static expression relationship, respectively, extracted from the scores for flute and harp (left) and percussions (right). The score is internally composed of 14 exercises which represents diverse combinations of musical articulations and structures and is reported in Figure 7: These can be grouped in two main types: 9 exercises mostly emphasizing a synchronicity in rhythm articulation (light gray) and 5 exercises mostly centered on synchronicity in melodic and expression articulation (dark gray). The musical stimuli have a duration of 3 minutes and a reference tempo of 112 BPM.

3.1.5. Procedure. Each duo had to perform the exercise, under six different conditions of emulated network delay, reported in Table 4. The minimum amount of latency, due to the two-way processing, was estimated in 28 ms, hence representing the first condition in Table 4. The sequence of six conditions was randomized for each duo. Before each session, each duo was briefed in Room 1, and the task was introduced, within the scope of InterMUSIC, without disclosing any information about the six network delay conditions. The score of the exercise was explained and handed out. Participants were informed about the duration of the exercise and the approximately overall duration of the experimental session (90 min) and introduced to the questionnaire on presence. They were asked to fill in the 5-item

questionnaire after each single repetition and the general 27-item questionnaire at the end of the whole session. Further comments were collected at the end of the test.

After the brief, the musicians settled in their respective room, and a 15-minute rehearsal was devoted to adjust their positioning, framing, and volume levels in order to provide a comfortable environment. In addition, they could rehearse and get acquainted with the score.

4. Results and Discussion

Being the experimental campaign for the pilot study limited to a reduced number of collected sessions ($S = 5$, for a total of 10 participating musicians), we provide the reader with a narrative of the information that we are able to extract from the analysis of the subjective and objective evaluations. In fact, we consider only three sessions out of five, that is, the couples C (percussions), D (harp/flute), and E (alto sax), since two sessions were either not fully completed or deeply biased (couples A, mandolins, and B, accordion/guitar). That being said, the analysis of the results shows the usefulness of the proposed framework as means to conceptualize and systematize the diverse aspects that affect the quality of a networked music learning scenario. The reduced sample size does not allow to stress any conclusive result; nonetheless, this pilot generated valuable methodological implications and hypotheses concerning the relevance of latency in NMP and the overall complexity at play.

4.1. Subjective Evaluation. The subjective evaluation is aimed at understanding the sense of presence in NMP performance and at providing a qualitative, yet reliable measure of NMP interaction and system, via presence constructs. In the current study, we consider the network latency as the main variable affecting the subjective experience of playing together in the networked space, that is, the focused concentration, the coherence, and immersion of the overall experience in the real and remote environments, in addition to the perceived quality of the performance.

The visual inspection of Table 2 returns a picture of an overall experience which is mostly perceived as puzzling, and yet intriguing. As it was expected, the musicians felt mostly neutral to the feeling of a place illusion (Q1.1 and Q1.3), and yet the sense of playing in the remote environment was experienced as sufficiently compelling (Q1.2). Despite the low effectiveness of the overall environment in generating a meaningful sharing experience, the musicians were able to concentrate on their performance (Q3.1). We hypothesize that if any sense of being together was felt, this was due to the inherent social characteristics of the music making task. Indeed, the overall quality of the display is perceived as poor (Q3.4), when not useless (Q3.3), which in turn affects the focused concentration on playing with the remote co-performer (Q4.1, Q4.2, and Q5.2).

Despite the distress, the musicians seemed to adjust to the experienced difficulties to a certain extent (Q5.1 and Q5.2) and make sense of the NMP interaction as a coherent whole (Q2.1). Again, we interpret this result rather as a goal-

FIGURE 6: Example of homorhythmic, unison melody, with static expression. Left: flute and harp. Right: wood blocks and tom-toms.

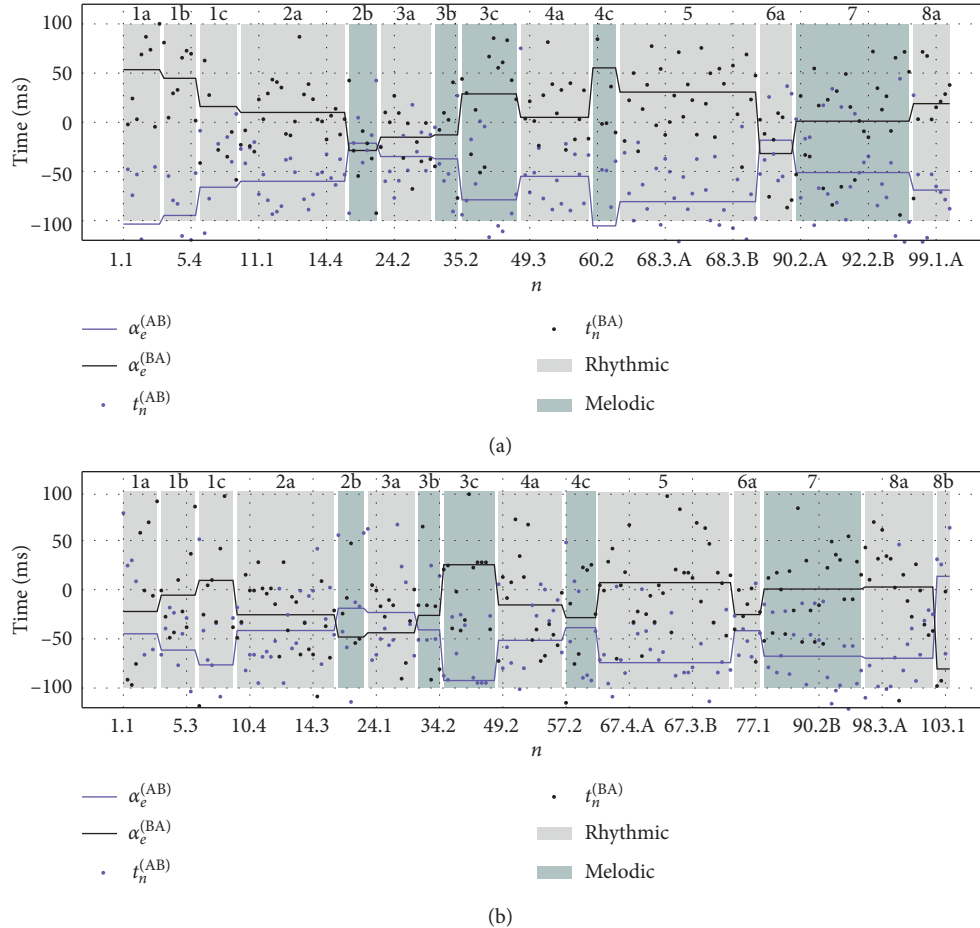


FIGURE 7: ISDs and asymmetry (averaged over each exercise) for couple (E). (a) Asymmetry for the couple E with 50 ms latency. (b) Asymmetry for the couple E with 67 ms latency.

TABLE 4: Latency values in the six conditions.

Condition	1	2	3	4	5	6
Delay (2-ways, ms)	28	33	50	67	80	134

directed quality of skilled musicians in managing to cope with the most adverse performing conditions. We can assume that such a situation is certainly not acceptable in a learning scenario. Nevertheless, the subjects proved their motivation and willingness to master the environment for the purpose of playing music remotely (Q5.2 and Q5.3).

In detail, Figure 8 shows the answers to the five postrepetition questions, regarding the focused concentration on the performance (Q1.2 and Q1.7, Figure 8(a) and Figure 8(b), respectively), the perceived coherence of the scenario (Q2.4, Figure 8(c)), the distraction factors (Q3.5,

Figure 8(d)), and the perceived quality of the performance (Q5.4, Figure 8(e)). The answer distribution highlights a negative effect of latency levels on the musicians involvement in the environment. However, these results must not be considered as conclusive; they rather highlight diverse inclinations and aptitudes of the subjects towards the delay issues. As general postexperiment comments indeed, participants C1 and D2 reported a lack of “musical connectedness,” despite the plausibility of the experience, which was also accounted by participants C2 and E1. It was reported that the decrease in involvement or flow, due to longer delays, increased the difficulty or impossibility to understand which was the cause of playing out of time. Of interest, this passage ended in an argument between the participants (couple E), whether the cause was ascribable to

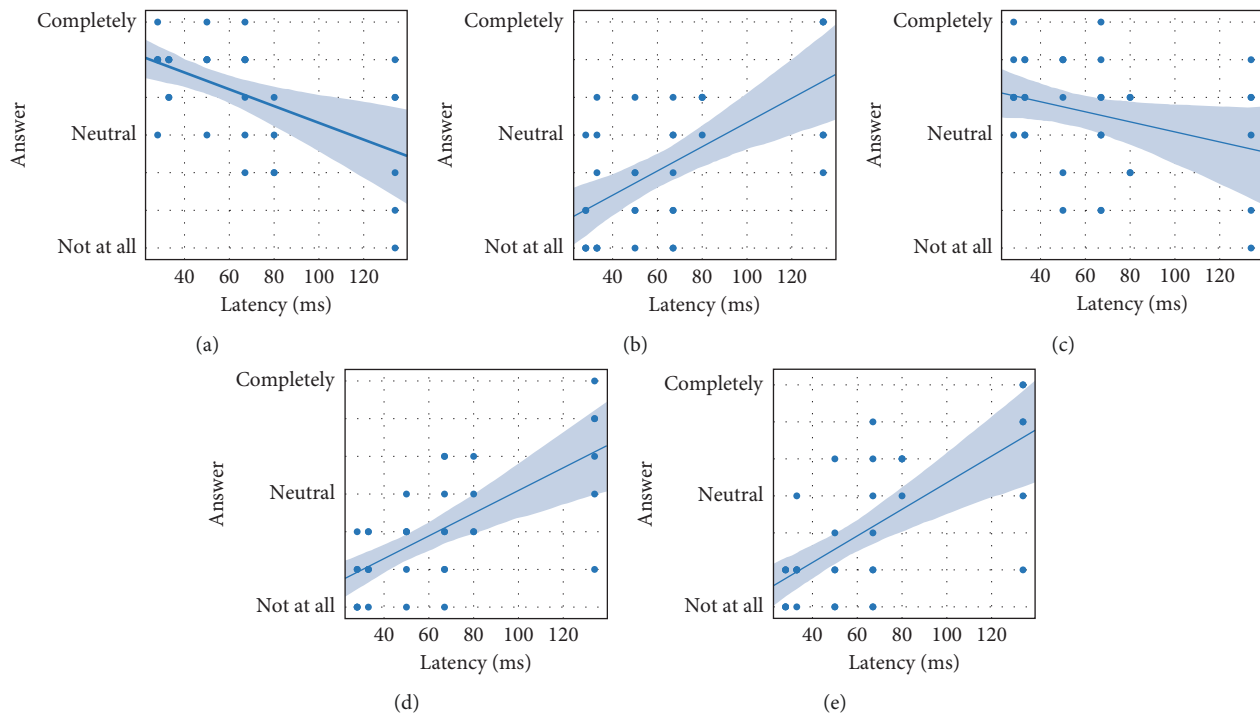


FIGURE 8: Answers to Q1.2, Q1.7, Q2.4, Q3.5, and Q5.4 in the postrepetition questionnaire with respect to the latency condition. Points represent individual answers to the questions. The solid straight lines represent the linear relationships, as obtained through linear regression, between the latency and the answers. The translucent band lines represent the 95% confidence interval for the regressions. (a) Answers to Q1.2. (b) Answers to Q1.7. (c) Answers to Q2.4. (d) Answers to Q3.5. (e) Answers to Q5.4.

the reduced commitment of the co-performer or the idiosyncrasy of the networked medium, thus reflecting a typical conflict and self-repair situation which can be compromised in telepresence systems [73]. As a final remark, it is interesting to look at the median values of the answers to the items on the interface awareness and quality (Section 3 of Table 2), which describe the perceived mastery of the interface, and are a subscale of the overall immersion of the system’s technology. The median values of the answers suggest that the musicians felt confident to concentrate on the musical task and make sense of the interface at hand. It is worth noting that the quality of the visual display does not seem to interfere or distract from performing (Q3.3), while the audio quality does (Q3.4). We hypothesize this is due to the poor quality of immersion of the visual representation (Q4.1), which makes it difficult for the performers to rely on vision to actively survey the performance (Q1.7). It is possible that this condition led the musicians to rather rely on the audio feedback in their performance. As a general comment, all the couples reported unanimously that the frontal screen resulted in a less natural interaction, as they normally use peripheral vision to monitor the co-performers on their side.

Taken together, the subjective evaluation of the NMP experience of the three duos returns a picture of a complex situation, wherein the issues at stake are multifaceted and systemic, especially with respect to the quality of the immersion and the sense of focused concentration awaited by musicians. From this viewpoint, a rather detailed reformulation of the

questions concerning the quality of the auditory and visual display would resolve the current, apparent ambiguity (e.g., the availability of certain information such as eye contact and breath attack, as a function of the system’s vividness and interactivity).

4.2. Objective Evaluation. In this section, we complement the subjective evaluation with the content analysis of the NMP performance. We show how measures of tempo trends can be used to interpret the performance strategies enacted by musicians to cope with NMP latency and in general to make sense of the medium behavior.

We compute the tempo trend $\bar{\delta}(n)$ for all the recordings of the experiments. In Figure 9, we show two sets of annotations and corresponding tempo trends (scatter plot), smoothed trend (continuous lines), and linear approximation computed from κ (dashed lines) for musicians in Room 1 (blue) and Room 2 (orange).

Figure 9(a) shows the tempo trend of the two percussionists (couple C), playing with a latency of 134 ms. From the visual inspection, it can be observed a smoothed and highly correlated trend showing a high degree of synchronization and increase of tempo during the performance. This behavior suggests that with such high latency, the musicians were not able to follow each other, opting instead to a master-slave approach. The percussionists’ performance was particularly challenging and severely hampered by the presence of a high audio feedback due to the nature of the instruments.

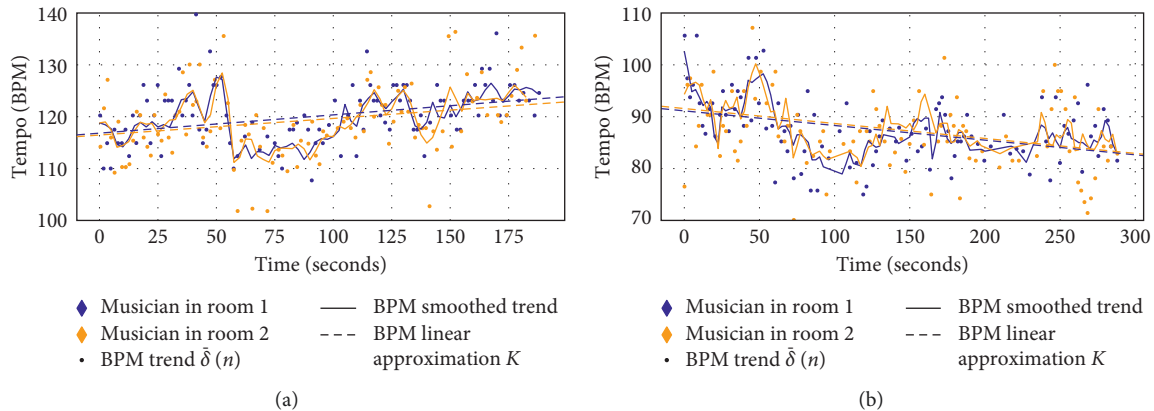


FIGURE 9: Tempo trend, in BPM, and its linear approximation for two sets of repetitions. (a) Tempo trend of the second repetition of couple C with 134 ms latency. (b) Tempo trend of the fifth repetition of couple D with 134 ms latency. (c) Legend of the figures.

A radically different situation is depicted in Figure 9(b), which shows the tempo trend of couple D performance (harp and flute), in the same latency condition of 134 ms; the two instrumentalists attempt to cope with the prohibitive latency condition by performing with a realistic interaction approach. The result, however, is a plain and progressive deceleration with respect to the reference tempo. In addition, in postexperiment debrief, the percussionist C2 commented that she mainly focused on keeping the internal tempo, while ignoring the co-performer's delayed performance. Both D1 and D2 confirmed that they were trying to follow each other's performance. The disparity of results can be accounted as well to the inherent difference between the instruments played by the two couples. Percussionists are effective and skilled followers, even at higher measures of delay [36], as it can be observed in the relative similar smooth fits of the tempo trend of both musicians, in Figure 9(a). D1, the harpist, and D2, the flautist, must confront greater challenges due to the constrictive relationship imposed by both the melodic and agogic constraints of the score. Unlike the percussionist, who has to focus mainly on the tactus, the harpist and the flautist have to preserve a synchronicity in pitch as well.

We also computed the ISDs $t_n^{(AB)}$ and $t_n^{(BA)}$ for all the repetitions and the asymmetry as the average ISDs over each of the 14 exercises in the score, which we will label as α_c . Figure 7 shows the performance asymmetry of couple E (alto sax), where the x -axis is the common beats n and the y -axis shows the ISDs (in milliseconds) and the two asymmetries. The gray-scale bands represent the 14 exercises, from 1a to 8a, whereas the dark gray areas indicate those exercises rather centered on the melodic and expression synchronicity. The profile in purple represents the asymmetry of musician E1.

From the visual inspection of Figure 7(a), which refers to the latency condition of 50 ms, a few observations can be made. Musician E1 constantly anticipates the beats and musician E2 constantly follows him, except in the two regions of exercises 2b and 6a. In particular, at exercise 7, E1 anticipates E2 on average by the exact amount of latency, and hence from his side E2 is playing exactly on time, suggesting the overall negotiation of master/slave approach within the duo [74].

At 67 ms of latency, the duo seems to follow another strategy. As shown in Figure 7(b), especially at the beginning of the performance, from 1a to 3b, both musicians are anticipating, in the attempt to cope with the latency. Conversely, in exercises 5, 7, and 8a, they return to the master-slave approach, with the same roles. This passage well represents the idiosyncrasy experienced by the musicians, in the attempt to understand the medium behavior, and provides a clue of the relevance of the overall coherence of a given scenario in providing consistent and undeviating responses. In general, the time needed by the couple to estimate the latency and opt for the most appropriate interaction strategy is a clue of a disruptive effect which, in the current NMP environment, prevents the musicians from making a reliable judgment and ascribe the mistakes or the poor quality of their performance to their acts or to the medium.

5. Conclusions

The early results of the pilot experiment, described in this paper, offer a picture of the many entangled aspects that characterize the user experience in networked environments for music interaction. The low sample size of the duos involved clearly prevents us from stressing any conclusive statement. Conversely, the experimental environment draws attention to the complexity of the many experiential and technological variables that affect the effectiveness of a network music performance. In this respect, the conceptual framework situated in the chamber music practice and learning scenario acts as a magnifying glass for observing the constructive elements that create the plausible illusion of playing and learning music in geographically displaced environments.

Despite the clearly limited number of cases, the research activity carried out so far has important methodological implications. First of all, the experience of chamber music making is put in the foreground, with respect to the general issue of presence in the virtual environment and to the technology behind the system that enables it. Secondly, the use of objective quality metrics is very helpful for exploring

design issues through in-action and on-action reflections [75]. The rationale of a well-designed score is that of considering the significant mediating elements of a classical chamber music performance, and yet retaining control on the musical structures implied in the relationships of synchronicity, which represents the major drawback of networked systems.

We described how pairs of musicians manage to adjust their interaction and negotiate the performance, even in the poor sensory conditions of a basic NMP staging (i.e., monoaural capturing and rendering and screen-based frontal view representation of the remote co-performer). The metric of asymmetry is a measure of misalignment between the performed parts, thus providing the link with the beat and the score (i.e., the chamber music scenario). As it has been shown, it provides a promising descriptor of the interaction approaches at play within the performance. The shapes emerging in Figure 7 reveal interesting hypotheses which are worth investigating and, namely, concerning the different effects of rhythm and melody/expression articulatory score indications on the musicians' negotiation of the networked performance. If this is the case, the pedagogical implications for the distance learning scenario may require the design of specific exercises to train pitch and temporal acuity to cope with adverse NMP conditions.

For this purpose, we plan to revise the score in order to balance the types of exercises. We should also reflect on the presentation order of the exercises, in terms of abrupt changes and homogeneity of types, yet without compromising the overall musical meaningfulness of the stimulus in the next experimental sessions. In this respect, an additional and valuable source of information is represented by the "44 Duos for Violin" by Béla Bartók, a series of pieces composed for pedagogical purposes, specifically addressed to train motor responses to aural problems, rhythmic and structural features, interpretation, and music memory [76].

On the other hand, the current experimental procedure actually reflects the management of the rehearsal, which represents the intermediate type of performance, between the concert and the lesson. The duos were given a short time to rehearse before the experiment; therefore, they essentially played by reading the score at first sight. Occurring mistakes certainly have an effect on the computed asymmetry. If we want to investigate the concert type in the networked space, it is important that duos be given the score to work with well in advance. Objective quality metrics will represent a more valuable resource to quantify the performance. In the same fashion, the subjective evaluation of the coherence of the NMP scenario is expected to be more grounded in expectations from the real world. The systematic inquiry of the rehearsal and lesson types, instead, may require different approaches, and ethnographic observations, protocol analysis, and objective metrics should be carried out over a more extended period of time. Future works are planned, where experiments regarding latency will be carried out with an extensive number of participants. We also aim at investigating more systematically the occurrence of place illusion in the NMP

music learning scenario, by experimenting more immersive audio-visual feedback solutions, such as binaural rendering and full-body projections. The presence questionnaire is also undergoing a substantial revision. For example, items referring to the quality of immersion, visual and auditory, are being detailed, based on the comments collected: chamber musicians make use of several visual and auditory signals to communicate in ensemble, such as foot tapping and breath attack, which should be made available and apparent by the system. Finally, we are introducing biometric measurement techniques, to make the use of the questionnaire more reliable [25, 50].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was conducted for the InterMUSIC project, which received the financial support of the Erasmus+ National Agency under the KA203 Strategic Partnership action under grant no. 2017-1-IT02-KA203-036770.

References

- [1] J. Lazzaro and J. Wawrzynek, "A case for network musical performance," in *Proceedings of the 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video, NOSSDAV '01*, pp. 157–166, ACM, New York, NY, USA, June 2001.
- [2] Á. Barbosa, "Displaced soundscapes: a survey of network systems for music and sonic art creation," *Leonardo Music Journal*, vol. 13, pp. 53–59, 2003.
- [3] L. Gabrielli and S. Squartini, *Wireless Networked Music Performance*, Springer, Berlin, Germany, 2016.
- [4] E. Lakiotakis, C. Liaskos, and X. Dimitropoulos, "Improving networked music performance systems using application-network collaboration," *Concurrency and Computation: Practice and Experience*, p. e4730, 2018.
- [5] C. Rottondi, C. Chafe, C. Allocchio, and A. Sarti, "An overview on networked music performance technologies," *IEEE Access*, vol. 4, pp. 8823–8843, 2016.
- [6] G. Hajdu, "Embodiment and disembodiment in networked music performance," in *Body, Sound and Space in Music and beyond; Multimodal Explorations*, C. Wöllner, Ed., pp. 257–278, Taylor & Francis, London, UK, 2017.
- [7] R. Mills, "Liminal worlds: presence and performer agency in tele-collaborative interaction," *Tele-Improvisation: Inter-cultural Interaction in the Online Global Music Jam Session*, pp. 145–166, Springer International Publishing, Cham, Switzerland, 2019.
- [8] F. Schroeder and P. Rebelo, "Sounding the network: the body as disturbant," *Leonardo Electronic Almanac*, vol. 16, no. 4-5, pp. 1–10, 2009.

- [9] J. Braasch, "The telematic music system: affordances for a new instrument to shape the music of tomorrow," *Contemporary Music Review*, vol. 28, no. 4-5, pp. 421-432, 2009.
- [10] F. Alpiste Penalba, T. Rojas-Rajs, P. Lorente, F. Iglesias, J. Fernández, and J. Monguet, "A telepresence learning environment for opera singing: distance lessons implementations over internet2," *Interactive Learning Environments*, vol. 21, no. 5, pp. 438-455, 2013.
- [11] M. Iorwerth, D. Moore, and D. Knox, "Challenges of using networked music performance in education," in *Proceedings of the UK 26th AES Conference*, vol. 8, Audio Education, August 2015.
- [12] K. Dye, "Student and instructor behaviors in online music lessons: an exploratory study," *International Journal of Music Education*, vol. 34, no. 2, pp. 161-170, 2016.
- [13] C. Johnson, "Teaching music online: changing pedagogical approach when moving to the online environment," *London Review of Education*, vol. 15, no. 3, pp. 439-456, 2017.
- [14] B. Kehrwald, "Democratic rationalisation on the network: social presence and human agency in networked learning," in *In Proceedings of the 7th International Conference on Networked Learning*, pp. 215-223, Aalborg, Denmark, May, 2010.
- [15] A. Margaryan, M. Bianco, and A. Littlejohn, "Instructional quality of massive open online courses (MOOCs)," *Computers & Education*, vol. 80, pp. 77-83, 2015.
- [16] C. Chafe and M. Gurevich, "Network time delay and ensemble accuracy: effects of latency, asymmetry," in *Proceedings of the 117th conference of the Audio Engineering Society Convention*, vol. 117, Audio Engineering Society, San Francisco, CA, USA, October 2004.
- [17] C. Chafe, J.-P. Cáceres, and M. Gurevich, "Effect of temporal separation on synchronization in rhythmic performance," *Perception*, vol. 39, no. 7, pp. 982-992, 2010.
- [18] S. Kolozali, M. Barthet, G. Fazekas, and M. Sandler, "Automatic ontology generation for musical instruments based on audio analysis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2207-2220, 2013.
- [19] C. Rottondi, M. Buccoli, M. Zanoni, D. Garao, G. Verticale, and A. Sarti, "Feature-based analysis of the effects of packet delay on networked musical interactions," *Journal of the Audio Engineering Society*, vol. 63, no. 11, pp. 864-875, 2015.
- [20] J. Forlizzi and K. Battarbee, "Understanding experience in interactive systems," in *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques, DIS'04*, pp. 261-268, ACM, New York, NY, USA, August 2004.
- [21] G. Novembre and P. E. Keller, "A conceptual review on action-perception coupling in the musicians' brain: what is it good for?," *Frontiers in Human Neuroscience*, vol. 8, p. 603, 2014.
- [22] P. Q. Pfordresher and C. Palmer, "Effects of hearing the past, present, or future during music performance," *Perception & Psychophysics*, vol. 68, no. 3, pp. 362-376, 2006.
- [23] R. I. Godøy and M. Leman, *Musical Gestures: Sound, Movement, and Meaning*, Routledge, New York, NY, USA, 2010.
- [24] T. Schubert, F. Friedmann, and H. Regenbrecht, "The experience of presence: factor analytic insights," *Presence: Teleoperators and Virtual Environments*, vol. 10, no. 3, pp. 266-281, 2001.
- [25] M. Meehan, S. Razzaque, M. C. Whitton, and F. P. Brooks, "Effect of latency on presence in stressful virtual environments," in *Proceedings of the IEEE Virtual Reality*, pp. 141-148, IEEE, Los Angeles, CA, USA, March 2003.
- [26] E. B. Nash, G. W. Edwards, J. A. Thompson, and W. Barfield, "A review of presence and performance in virtual environments," *International Journal of Human-Computer Interaction*, vol. 12, no. 1, pp. 1-41, 2000.
- [27] W. Woszczyk, J. Cooperstock, J. Roston, and W. Martens, "Shake, rattle, and roll: getting immersed in multisensory, interactive music via broadband networks," *Journal of the Audio Engineering Society*, vol. 53, no. 4, pp. 336-344, 2005.
- [28] R. Nordahl and N. C. Nilsson, *The Sound of BeingThere Presence and Interactive Audio in Immersive Virtual Reality*, Oxford University Press, Oxford, UK, 2014.
- [29] R. Skarbez, F. P. Brooks Jr., and M. C. Whitton, "A survey of presence and related concepts," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, p. 96, 2017.
- [30] T. W. Schubert, "A new conception of spatial presence: once again, with feeling," *Communication Theory*, vol. 19, no. 2, pp. 161-187, 2009.
- [31] M. Leman, P.-J. Maes, L. Nijs, and E. Van Dyck, "What is embodied music cognition?," in *Springer Handbook of Systematic Musicology*, R. Bader, Ed., pp. 747-760, Springer, Berlin, Heidelberg, 2018.
- [32] C. Sora, S. Jordà, and L. Codina, "Chasing real-time interaction in new media: towards a new theoretical approach and definition," *Digital Creativity*, vol. 28, no. 3, pp. 196-205, 2017.
- [33] S. Duffy, P. G. Healey et al., "Co-ordinating non-mutual realities: the asymmetric impact of delay on video-mediated music lessons," in *Proceedings of the 39th Annual Conference of the Cognitive Science Society*, London, UK, July 2017.
- [34] C. Chafe, M. Gurevich, G. Leslie, and S. Tyan, "Effect of time delay on ensemble accuracy," in *Proceedings of the International Symposium on Musical Acoustics*, vol. 31, p. 46, Nara, Japan, March-April 2004.
- [35] G. Aschersleben, "Temporal control of movements in sensorimotor synchronization," *Brain and Cognition*, vol. 48, no. 1, pp. 66-79, 2002.
- [36] J. R. Cooperstock, "Interacting in shared reality," in *Proceedings of the 11th International Conference on Human-Computer Interaction*, Las Vegas, NV, USA, July 2005.
- [37] C. Knudsen, "Synchronous virtual spaces-transparent technology for producing a sense of presence at a distance," in *Proceedings of the 2001 Telecommunications for Education and Training Conference, TET*, Charles University Prague, Prague, Czech Republic, May 2001.
- [38] C. Hendrix and W. Barfield, "The sense of presence within auditory virtual environments," *Presence: Teleoperators and Virtual Environments*, vol. 5, no. 3, pp. 290-301, 1996.
- [39] M. Gurevich, D. Donohoe, and S. Bertet, "Ambisonic spatialization for networked music performance," in *Proceedings of the 17th International Conference on Auditory Display (ICAD2011)*, Budapest, Hungary, June 2011.
- [40] S. Farner, A. Solvang, A. Sæbo, and U. P. Svensson, "Ensemble hand-clapping experiments under the influence of delay and various acoustic environments," *Journal of the Audio Engineering Society*, vol. 57, no. 12, pp. 1028-1041, 2009.
- [41] J. Malloch and M. M. Wanderley, "Embodied cognition and digital musical instruments: design and performance," in *The Routledge Companion to Embodied Music Interaction*, M. Lesaffre, P.-J. Maes, and Leman, Eds., pp. 438-447, Routledge, Abingon, UK, 2017.
- [42] C. Chafe, "I am streaming in a room," *Frontiers in Digital Humanities*, vol. 5, no. 27, 2018.
- [43] Y. Iwaya and B. F. Katz, "Distributed signal processing architecture for real-time convolution of 3d audio rendering for

- mobile applications,” in *Proceedings of the International Conference on Virtual Reality and Augmented Reality*, pp. 148–157, Springer, London, UK, October 2018.
- [44] B. Boren and M. Musick, “Spatial organization in musical form,” in *Proceedings of the International Computer Music Conference*, pp. 33–38, ICMC, Daegu, South Korea, August 2018.
- [45] J.-P. Cáceres and C. Chafe, “Jacktrip: under the hood of an engine for network audio,” *Journal of New Music Research*, vol. 39, no. 3, pp. 183–187, 2010.
- [46] C. Drioli, C. Allocchio, and N. Buso, “Networked performances and natural interaction via LOLA: low latency high quality A/V streaming system,” *Information Technologies for Performing Arts, Media Access, and Entertainment*, pp. 240–250, Springer, Berlin, Germany, 2013.
- [47] P. Holub, J. Matela, M. Pulec, and M. Šrom, “Ultragrid: low-latency high-quality video transmissions on commodity hardware,” in *Proceedings of the 20th ACM International Conference on Multimedia*, pp. 1457–1460, ACM, New York, NY, USA, October–November 2012.
- [48] J. V. Draper, D. B. Kaber, and J. M. Usher, “Telepresence,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 40, no. 3, pp. 354–375, 1998.
- [49] J. J. Cummings and J. N. Bailenson, “How immersive is enough? A meta-analysis of the effect of immersive technology on user presence,” *Media Psychology*, vol. 19, no. 2, pp. 272–309, 2016.
- [50] M. Slater, “How colorful was your day? Why questionnaires cannot assess presence in virtual environments,” *Presence: Teleoperators and Virtual Environments*, vol. 13, no. 4, pp. 484–493, 2004.
- [51] M. Slater, “Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1535, pp. 3549–3557, 2009.
- [52] M. Leman, *Embodied Music Cognition and Mediation Technology*, Mit Press, Cambridge, MA, USA, 2007.
- [53] J. W. Davidson and J. M. M. Good, “Social and musical coordination between members of a string quartet: an exploratory study,” *Psychology of Music*, vol. 30, no. 2, pp. 186–201, 2002.
- [54] D. Glowinski, G. Gnecco, S. Piana, and A. Camurri, “Expressive non-verbal interaction in string quartet,” in *Proceedings of the Humaine Association Conference on Affective Computing and Intelligent Interaction*, pp. 233–238, IEEE, Geneva, Switzerland, September 2013.
- [55] A. Chirico, S. Serino, P. Cipresso, A. Gaggioli, and G. Riva, “When music “flows”. state and trait in musical performance, composition and listening: a systematic review,” *Frontiers in Psychology*, vol. 6, p. 906, 2015.
- [56] B. G. Witmer and M. J. Singer, “Measuring presence in virtual environments: a presence questionnaire,” *Presence: Teleoperators and Virtual Environments*, vol. 7, no. 3, pp. 225–240, 1998.
- [57] L. Comanducci, M. Buccoli, M. Zanoni et al., “Investigating networked music performances in pedagogical scenarios for the intermusic project,” in *Proceedings of the 23rd Conference of Open Innovations Association (FRUCT)*, pp. 119–127, IEEE, Bologna, Italy, November 2018.
- [58] L. Badino, A. D’Ausilio, D. Glowinski, A. Camurri, and L. Fadiga, “Sensorimotor communication in professional quartets,” *Neuropsychologia*, vol. 55, pp. 98–104, 2014.
- [59] C. Alexandraki and R. Bader, “Anticipatory networked communications for live musical interactions of acoustic instruments,” *Journal of New Music Research*, vol. 45, no. 1, pp. 68–85, 2016.
- [60] S. Vandemoortele, K. Feyaerts, G. De Bièvre, M. Reybrouck, G. Brône, and T. De Baets, “Gazing at the partner in musical trios: a mobile eye-tracking study,” *Journal of Eye Movement Research*, vol. 11, no. 2, p. 6, 2018.
- [61] M. Yoshie, K. Kudo, T. Murakoshi, and T. Ohtsuki, “Music performance anxiety in skilled pianists: effects of social-evaluative performance situation on subjective, autonomic, and electromyographic reactions,” *Experimental Brain Research*, vol. 199, no. 2, pp. 117–126, 2009.
- [62] S. Duffy and P. Healey, “A new medium for remote music tuition,” *Journal of Music, Technology and Education*, vol. 10, no. 1, pp. 5–29, 2017.
- [63] A. Williamon and J. W. Davidson, “Exploring co-performer communication,” *Musicae Scientiae*, vol. 6, no. 1, pp. 53–72, 2002.
- [64] E. Geelhoed, D. Prior, and M. Rofe, “Designing a system for online orchestra: microphone evaluation and cost-benefit analysis,” *Journal of Music, Technology and Education*, vol. 10, no. 2-3, pp. 213–230, 2017.
- [65] B. Boren and A. Genovese, “Acoustics of virtually coupled performance spaces,” in *Proceedings of the 24th International Conference on Auditory Display-ICAD2018*, pp. 80–86, Michigan Technological University, Houghton, MI, USA, June 2018.
- [66] D. Markovic, F. Antonacci, A. Sarti, and S. Tubaro, “Soundfield imaging in the ray space,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 12, pp. 2493–2505, 2013.
- [67] L. Bianchi, F. Antonacci, A. Sarti, and S. Tubaro, “Model-based acoustic rendering based on plane wave decomposition,” *Applied Acoustics*, vol. 104, pp. 127–134, 2016.
- [68] S. Serafin, M. Geronazzo, C. Erkut, N. C. Nilsson, and R. Nordahl, “Sonic interactions in virtual reality: state of the art, current challenges, and future directions,” *IEEE Computer Graphics and Applications*, vol. 38, no. 2, pp. 31–43, 2018.
- [69] S. Delle Monache, M. Buccoli, L. Comanducci et al., “Time is not on my side: network latency, presence and performance in remote music interaction,” in *Proceedings of the XXII CIM Colloquium on Music Informatics-Machine Sounds, Sound Machines*, pp. 152–158, Udine, Italy, November 2018.
- [70] B. G. Witmer, C. J. Jerome, and M. J. Singer, “The factor structure of the presence questionnaire,” *Presence: Teleoperators and Virtual Environments*, vol. 14, no. 3, pp. 298–312, 2005.
- [71] G. Robillard, S. Bouchard, P. Renaud, and L. Cournoyer, “Validation canadienne-française de deux mesures importantes en réalité virtuelle: l’immersivité et le questionnaire de présence,” in *Proceedings of the 25e Congrès Annuel de la Société Québécoise pour la Recherche en Psychologie (SQRP)*, Trois-Rivières, Canada, November 2002.
- [72] B. Bartók, *Mikrokosmos, 153 Progressive Piano Pieces, New Definitive Edition*, Boosey & Hawkes Music Publishers Limited, London, UK, 1987.
- [73] S. Duffy and P. Healey, “Using music as a turn in conversation in a lesson,” in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 35, pp. 2231–2236, Queen Mary University of London, London, UK, August 2013.
- [74] E. Carôt and C. Werner, “Network music performance-problems, approaches and perspectives,” in *Proceedings of the Music in the Global Village Conference*, vol. 162, pp. 23–10, Budapest, Hungary, September 2007.

- [75] D. A. Schön, *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, New York, NY, USA, 1983.
- [76] S. M. Genovefa, "The pedagogical significance of the Bartók duos," *American String Teacher*, vol. 12, no. 3, pp. 22–29, 1962.

Research Article

O2: A Network Protocol for Music Systems

Roger B. Dannenberg 

School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Correspondence should be addressed to Roger B. Dannenberg; rbd@cs.cmu.edu

Received 2 January 2019; Revised 18 March 2019; Accepted 4 April 2019; Published 6 May 2019

Guest Editor: Federico Fontana

Copyright © 2019 Roger B. Dannenberg. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

O2 is a communication protocol for music systems that extends and interoperates with the popular Open Sound Control (OSC) protocol. Many computer musicians routinely deal with problems of interconnection, unreliable message delivery, and clock synchronization. O2 solves these problems, offering named services, automatic network address discovery, clock synchronization, and a reliable message delivery option, as well as interoperability with existing OSC libraries and applications. Aside from these new features, O2 owes much of its design to OSC, making it easy to migrate existing OSC applications to O2 or for developers familiar with OSC to begin using O2. O2 addresses the problems of interprocess communication within distributed music applications.

1. Introduction

Music software and other artistic applications of computers are often organized as a collection of communicating processes. Simple protocols such as MIDI [1] and Open Sound Control (OSC) [2] have been very effective for this, allowing users to piece together systems in a modular fashion. Shared communication protocols allow implementers to use a variety of languages, apply off-the-shelf applications and devices, and interface with low-cost sensors and actuators. In addition, mobile applications intrinsically run on multiple distributed host computers and require a communication protocol for any kind of coordination.

Figure 1 illustrates several common organizations for networked music systems. Figure 1(a) shows the “input/mapper/output” structure, where sensors and input devices stream values to a control system that maps sensor values to control parameters, which are passed on to a music synthesizer or audio signal processor [3]. Examples of this approach include SensorChimes [4] and play-along mappings of Fiebrink *et al.* [5]. The libmapper system is a communication protocol designed to support this approach [6].

Figure 1(b) illustrates a “conductor/ensemble” structure commonly used in laptop orchestra and mobile device music systems. Multiple performers can join and leave the ensemble by connecting to a central conductor or coordinator that directs the performers. Examples are described by Essl [7],

Trueman *et al.* [8], and Dannenberg *et al.* [9]. This same configuration is used in wide-area networked music performances on a global scale such as quintet.net [10] and the Global Network Orchestra [11].

Figure 1(c) illustrates a peer-to-peer organization that is used in networked performances characterized by autonomy and emergent behavior as opposed to the top-down control seen in the conductor/ensemble model. Gresham-Lancaster offers an interesting early history and discussion of this approach [12]. More examples of all three structures are described by Wright [13].

In all of these patterns, we see networking has advanced from point-to-point communication to more flexible and comprehensive communication substrates often used as much for software modularity and resilience as for communication. We introduce the protocol, O2, that provides for communication and coordination among music processes and offers some important new features over previous protocols such as OSC.

A common problem in any distributed system is how to initialize connections. For example, typical OSC servers do not have fixed IP addresses and cannot be found via DNS servers as is common with Web servers. Instead, OSC users usually enter IP addresses and port numbers manually. The numbers cannot be “compiled in” to code because IP addresses are dynamically assigned and could change between development, testing, and performance. O2

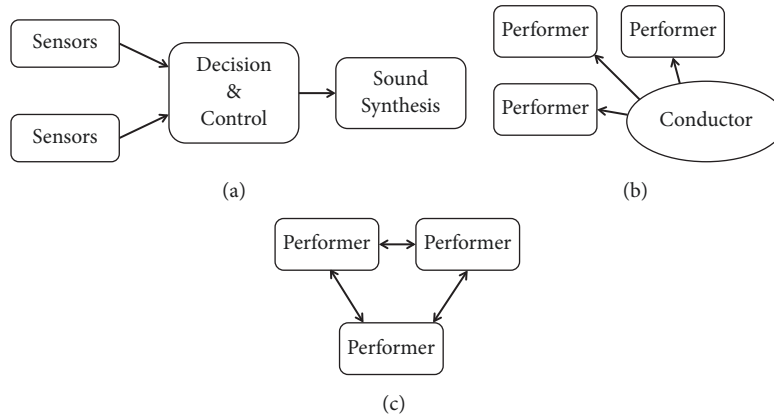


FIGURE 1: Common structures for networked music systems: (a) the “input/mapper/output” structure; (b) the “conductor/ensemble” structure; (c) peer-to-peer structure.

eliminates most network configuration problems, allowing programmers to create and address services with fixed, human-readable names.

Music applications often have two conflicting requirements for message delivery. Sampled sensor data should be sent with minimum latency. Lost data is of little consequence since a new sensor reading will soon follow, and retransmitting stale sensor data serves little purpose. This calls for a best-effort delivery mechanism such as UDP. On the other hand, some messages are critical and one-time only, e.g., “stop now.” These critical messages are best sent with a reliable delivery mechanism such as TCP.

Another desirable feature is timed message delivery, especially for music where timing is critical. One powerful method of reducing timing jitter in networks is to pre-compute commands and send them in advance for precise delivery according to timestamps. O2 facilitates this forward-synchronous approach [14] with timestamps and clocks.

Our goal has been to create a simple, extensible communication mechanism for modern computer music (and other) systems. O2 is inspired by OSC, but there are some important differences. While OSC does not specify details of the transport mechanism, O2 uses TCP and UDP over IP (which in turn can use Ethernet, WiFi, and other data link layers). By assuming a common IP transport layer, it is relatively straightforward to add discovery, a reliable message option, and accurate timing.

In the following section, we describe important and novel features of O2. This is followed by a section on related work. Then, we describe the design and implementation, focusing on novel aspects of O2. A communication protocol is mainly useful as “glue” between different systems. In the section “Interoperation,” we describe how O2 interoperates with Open Sound Control, Web applications, and various languages. Finally, we present some performance measurements, a summary, and conclusions.

2. O2 Features and API

The main organization of O2 is illustrated in Figure 2. We will first introduce some O2 terminology. An O2 *host* is a

computer with an IP address. An O2 *process* is a running program. There may be multiple processes running on a single host. An O2 *ensemble* is a named collection of *processes* that communicate and share services. Every O2 process belongs to one and only one ensemble. This allows multiple independent performers or groups to use O2 over the same network without interference. An O2 *address* is a URL-like string designating a function or method. For example, `/synth/filter/cutoff` might address a function to set the filter cutoff frequency in the `synth` service.

2.1. Creating a Service. The top-level (first) node in an *address* names an O2 *service*. A service is an abstraction for a set of functions or a resource such as a synthesizer, a display, a sensor, or a controller. A service is accessible via O2 *messages*, which consist of an *address* and a set of typed parameters. Services can be created dynamically by any process, services are “owned” by a process and automatically discovered by all other O2 processes, and all messages addressed to a service are delivered by invoking a registered callback function within the process.

To create a service, one writes

```
o2_initialize(ensemble); // one-time O2
startup
o2_service_new(service); // create a new
service
```

where *ensemble* is a unique ensemble name.

Typically, each full address in the hierarchical name space represents a function. To associate a function with an O2 address, call

```
o2_method_new(address, types, handler,
info, coerce, parse);
```

where *address* is the full address, e.g. `/synth/filter/cutoff`, *types* gives the expected parameter types (for example, “si” specifies that a string and a 32-bit integer parameter are expected), *handler* is the address of the callback function to which the parameters are passed, and *info* is an additional parameter to pass to this handler function.

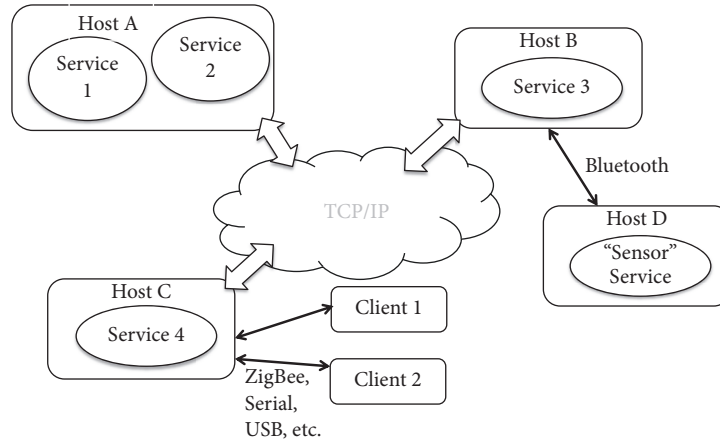


FIGURE 2: A distributed O2 ensemble showing processes connected by TCP/IP (wireless and/or wired) over a local area network, running multiple services, with additional single-hop links over Bluetooth, ZigBee, etc. Services on Host A may run within a single process or in separate processes, and all processes may act as clients, sending messages to any service.

The *coerce* and *parse* parameters give additional control over message handling.

2.2. Discovery. Services are automatically detected and connected by O2. This solves the problem of manually entering IP addresses and port numbers. The discovery process is detailed in Section 5.

Messages in O2 are delivered only when the address exists, so if a service has not been created, a network connection is lost, or a process terminates, the message will be dropped without raising any exceptions. This is often a great simplification for applications. For example, a sensor process can send sensor data to a consumer service whether or not the service exists. It is not necessary to carefully start the server process before starting the client (sensor process). When the service is active, it gets the data; when it is not, the service lookup will fail locally, and no message will be sent. In practice, this can cause problems when a client needs to configure a service or request important information from it. In these cases, it is common for clients to call

```
o2_status(service)
```

until the *service* is created and discovered, at which point communication can begin. It might be noted that similar problems arise even with simple client-server communication with TCP/IP: the client's *connect* call will fail (after at least a round-trip across the network) if the server does not exist.

2.3. Sending Messages. Messages can be sent either with lowest latency or reliably using two different functions:

```
o2_send(address, time, types, val1,
        val2, ...);
o2_send_cmd(address, time, types, val1,
            val2, ...);
```

where *types* (in the C implementation) specifies the types of parameters. The first form (*o2_send*) uses UDP, which is

most common for OSC, and the second form (*o2_send_cmd*) sends a “command” using TCP, ensuring that the message will be delivered.

2.4. Timed Message Delivery. O2 runs a clock synchronization service to establish a shared clock across the distributed ensemble. The master clock is provided to O2 by calling

```
o2_set_clock(clock_callback_fn, info);
```

where *clock_callback_fn* is a function pointer that provides a time reference and *info* is a parameter to pass to the function. The master clock can be the local system time of some host, an audio sample count converted into seconds (for synchronizing to audio), SMPTE time code, GPS, or any other time reference. Notice that every send operation (*o2_send*, *o2_send_cmd*) specifies a delivery time. A send operation always transmits a message immediately to the receiver without regard for the timestamp. If the message arrives early, it is held in a priority queue until the delivery time. Thus, if messages can be sent early (e.g., by increasing overall latency), message delivery times can be very precise (reducing timing jitter) [14]. This is often important for accurate timing in music.

3. Related Work

Table 1 summarizes some properties and features of various systems for networked music applications. Many simple systems implement application-specific protocols using TCP or UDP to carry text or binary data. Open Sound Control (OSC) offers a simple but very successful standard protocol for a variety of music and media applications [2]. The protocol is extensible and supported by many systems and implementations. The basic design supports a hierarchical address space of variables that can be set to typed values using messages. The messages can convey multiple values, and thus OSC may be viewed as a remote function or method invocation protocol. One very appealing quality of OSC, as compared

TABLE 1: Summary of properties of O2 and some alternative communication systems for real-time music networks. “Location Transparency” means that processes or services can be addressed by name or function rather than by direct IP addresses and/or port numbers. “Data Streams” refers to libmapper’s unique ability connect producers of numerical data streams to consumers, including mapping and filtering options to adapt sensor outputs to controller system inputs.

	Location Transparency	Typed, Named Parameters	Discovery	Mixed Reliable & Best Effort	Timed Delivery	Data Streams	Comments
TCP or UDP							
OSC		✓					Timed delivery is possible; rarely implemented.
O2	✓	✓	✓	✓	✓		
libmapper	✓	✓	✓			✓	
Landini	✓	✓	✓	✓	✓		Messages sent indirectly through local servers; N^2 ping traffic limits ensemble size.

to distributed object systems (such as CORBA [15]), is its simplicity. In particular, the OSC address space is text-based and similar to a URL, which means that programmers can write human-readable addresses directly without the need for interface description languages or preprocessors to translate strings to binary codes. It has been argued that OSC would be more efficient if it used fixed-length binary addresses, but the success of OSC suggests that users are not interested in greater efficiency at the cost of more complexity.

Discovery in O2 automatically shares IP addresses and port numbers to establish connections between processes. The liboscqs (<http://liboscqs.sourceforge.net>) and OSCgroups (<http://www.rossbencina.com/code/oscgroups>) library and osctools (<https://sourceforge.net/projects/osctools>) project support discovery through Zeroconf [16] and other systems. Malloch [6] describes the use of multicast for discovery, but this requires an agreed-upon and reserved multicast address. Essl [7] advocates the use of Bonjour (Apple’s implementation of Zeroconf), and included Bonjour-based discovery into networked mobile-phone-based music software. Bonjour has been slow to become a standard cross-platform service, but it offers a good solution to discovery. Eales and Foss explored discovery protocols in connection with OSC for audio control [17]; however their emphasis is on querying the structure of an OSC address space rather than discovery of servers on the network.

LANdini [18] addresses many of the problems that O2 is designed to solve. To solve the problem of discovery, LANdini runs a server on each host, and servers discover other servers using UDP broadcast messages. A sending process delivers a message to the local LANdini server, which forwards the message to the destination host’s LANdini server, and from there the message is forwarded again to the receiving process. This means that three messages are sent for each

application-level message delivery. Since LANdini is built using OSC, which in turn uses UDP, LANdini implements its own retransmission scheme for reliable message sending. An implementation with n hosts sends $3n(n-1)$ messages per second, limiting the practical ensemble size. LANdini also performs clock synchronization among servers, but there is no additional synchronization between servers and the ultimate destination processes.

The libmapper system [6] is particularly aimed at “input/mapper/output” systems (Figure 1(b)) and directly supports connections with specified mappings and filters, which is beyond the scope of O2. However, libmapper seems to be less suited to more general communication including over wide area network systems.

Software developers have also discussed and implemented OSC over TCP for reliable delivery. Systems such as liblo (<http://liblo.sourceforge.net/>) offer either UDP or TCP, but not both unless multiple servers are set up, one for each protocol.

Clock synchronization techniques are widely known. Cristian [19] described a simple method that is the basis for clock synchronization in O2. Madgwick *et al.* [20] describe a method for OSC that uses broadcast from a master and assumes bounds on clock drift rates. Brandt and Dannenberg describe a round-trip method with proportional-integral controller [14]. OSC itself supports timestamps, but only in message bundles, and there is no built-in clock synchronization.

4. Design Considerations and Details

In designing O2, we considered that computing technology is not as limited today as it was when OSC was designed. In particular, embedded computers running Linux or otherwise supporting TCP/IP are now small and inexpensive, and

the Internet of Things (IOT) will spur further development of low-cost, low-power, networked sensors and controllers. While OSC deliberately avoided dependency on a particular transport technology, enabling low-cost, lightweight communication, O2 assumes that TCP/IP is available to (most) hosts. O2 uses that assumption to offer new features. We also use floating point to simplify clock synchronization calculations because floating point hardware has become commonplace even on low-cost microcontrollers, or at least microcontrollers are fast enough to emulate floating point as needed.

4.1. Addresses in O2. In OSC, most applications require users to manually set up connections by entering IP and port numbers. In contrast, O2 provides “services.” An O2 *service* is just a unique name used to route messages within a distributed application. O2 addresses begin with the service name, making services the top-level node of a global address space. Thus, while OSC might direct a message to `/filter/cutoff` at IP 128.2.1.39, port 3, a complete O2 address would be written simply as `/synth/filter/cutoff`, where `synth` is the service name.

4.2. UDP versus TCP for Message Delivery. The two main protocols for delivering data over IP are TCP and UDP. TCP is “reliable” in that messages are retransmitted until they are successfully received, and subsequent messages are queued to insure in-order delivery. UDP messages are often more appropriate for real-time sensor data because new data can be delivered immediately rather than waiting for delivery or even retransmission of older data. O2 supports both protocols.

To illustrate the need for both delivery protocols, we wrote simple O2 programs to send 20,000 short messages at 20 messages per second, alternating use of TCP and UDP between two personal computers sharing a local WiFi network. Five of 10,000 UDP messages (0.05%) were lost by the network, and the maximum delay between receive times of two consecutive UDP messages was 343 ms. Of course, all TCP messages were delivered, and the maximum delay between messages was also 343 ms. However, the delay between TCP messages was greater than 110 ms 303 times (3%) but only 89 times (0.9%) with UDP. Thus, TCP retransmissions generate a significant number of delays that might be avoided using UDP when packet loss is not critical. These numbers are highly dependent upon network behavior, but it is clear that TCP and UDP are both useful.

4.3. Time Stamps and Synchronization. O2 protocols include clock synchronization and time-stamped messages. Unlike OSC, *every* message is time-stamped, but one can always send 0.0 to mean “as soon as possible.” Synchronization is initiated by clients, which communicate independently with a designated master clock process.

4.4. Taps and Debugging Support. Debugging a distributed application is difficult in part because there is no single point of control. When a component fails to behave as expected, it is helpful to know what messages, if any, are being received there. O2 has a message monitoring facility:

```
o2_tap(tappee, tapper);
```

installs a “tap,” where messages delivered to service *tappee* (a string) are copied, the message address is modified by replacing *tappee* with *tapper*, and the new message is delivered (to service *tapper*). This facility supports the construction of a remote message monitoring program. A simple monitor has been implemented and is described below in the “Interoperation” section.

5. Implementation

The O2 implementation is small and leverages existing functionality in TCP/IP. The OS X library, for example, is about 130KB, compared to a popular OSC library, `liblo`, which is 100KB. In this section, we describe the implementation of the important new features of O2.

5.1. Service Discovery. To send a message, an O2 client must map the service name from the address (or address pattern) to an IP address and port number. We considered existing discovery protocols such as ZeroConf (also known as Bonjour, Rendezvous, and Avahi) but decided a simpler protocol based on UDP broadcast messages would be smaller, more portable to small systems, and give more flexibility if new requirements arise. In particular, ZeroConf must be installed and configured as a server on some operating systems, whereas discovery in O2 is integrated with the O2 library.

Figure 3 illustrates the discovery sequence. When an O2 process is initialized, it allocates a server port and broadcasts its server port, host IP address, and an ensemble name. Any process running an instance of O2 with the same ensemble name will receive one of these broadcasts, establish a TCP connection to the remote process, and exchange service names. Multiple independent ensembles can share the same local area network without interference if they have different ensemble names. O2 retransmits discovery information every *discovery period* since there is no guarantee that all processes receive the first transmissions. The discovery period is short enough to avoid long discovery latency and long enough to avoid too much network traffic. (See “Scaling Issues” below.)

To direct a message to a service, the sender extracts the service name from the full address and uses a hash table to find an entry for the service name. The entry contains the corresponding TCP socket or address for a UDP message. The message is then sent to the process. The receiver uses another hash table to find a handler for the message and invokes the handler.

5.2. Discovery Ports. As described above, each O2 *process* performs discovery directly without relying on a separate server process. Unfortunately, this requires the use of multiple ports (one per O2 process). In an early implementation, it was thought that each process would allocate 1 of n predefined discovery port numbers. Then, discovery messages would be broadcast to all n port numbers. Since message traffic grows linearly with n , there is pressure to keep n small, but n is the upper bound on the number of processes that can run on a single host, so it seems that n should be at least 10, increasing discovery message traffic by a factor of $n \geq 10$.

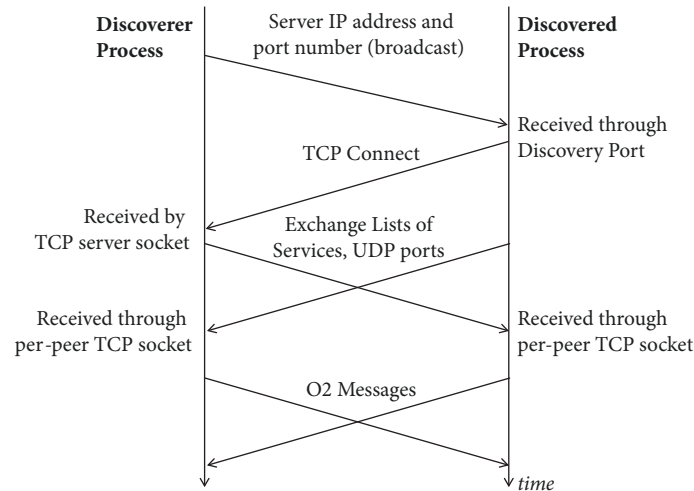


FIGURE 3: Discovery protocol. After receiving an O2 process’s IP address and port number, a peer establishes a TCP connection and the peers exchange service names and UDP ports (for messages sent via UDP), after which processes can exchange O2 messages. There is a chance in a fully symmetrical protocol that each process can connect to the other simultaneously. To avoid this, the TCP Connect is only issued from the process with the lower IP:port combination, breaking the symmetry.

Perhaps surprisingly, we can make n arbitrarily large without necessarily increasing discovery message traffic, using the following method. Discovery port numbers are ordered, and processes allocate the first available port from an ordered list: $port_1, port_2, \dots, port_n$. Now, consider two processes A and B, which have allocated $port_{PA}$ and $port_{PB}$. Process A will only broadcast to ports $1 \dots PA$, and process B will only broadcast to ports $1 \dots PB$. One of the following must be true: either $PA < PB$, $PA > PB$, or $PA = PB$. If $PA < PB$, then B will broadcast to $port_{PA}$. If $PA > PB$, then A will broadcast to $port_{PB}$. If $PA = PB$, then A will broadcast to $port_{PB}$ and B will broadcast to $port_{PA}$. In all cases, a discovery message will be sent from one process to the other, and both will be connected. If A and B are on separate hosts, each will typically open $port_1$ and only broadcast to $port_1$. In the case of $m \leq n$ processes on the same host, they will typically open $port_1$ through $port_m$ requiring $m(m+1)/2$ broadcast messages every period, but we do not expect m to be large.

The default O2 configuration sets $n = 16$, allowing up to 16 O2 processes on one host. Ports are not reserved through the Internet Assigned Numbers Authority (IANA), but since O2 discovery will work as long as *any* port from the set of 16 is available, reserved ports are not critical. In summary, the advantage of our port allocation and discovery scheme is that rather than broadcasting from m processes to every possible discovery port, costing $16m$ messages, a distributed ensemble with 1 process on each of m hosts will broadcast only m messages per discovery period. Furthermore, this peer-to-peer system does not require a separate discovery server process.

Unfortunately, not all networks allow broadcast messages, and broadcasts are (usually) only available within a local area network. As an alternative to discovery, any O2 process can find all services by calling

```
o2_hub(ip_address, port) ;
```

which provides the IP address and port for *one* other O2 process. O2 then contacts the other process and receives a list of the current services as well as any future updates. A typical use case is a wireless network where broadcast is disabled and IP addresses are assigned dynamically, making it impossible to use fixed IP addresses in the application software. Instead, a designated “hub” O2 process posts its IP address and port to a simple web server. Other O2 processes request the hub information from the server and then call `o2_hub` to form a fully connected ensemble. Posting an address to a server in this way is performing a function similar to DNS. An alternative to creating this specialized web service is to configure hosts to use a dynamic DNS service and have at least one O2 process register its address there.

5.3. Timestamps and Clock Synchronization. O2 implements clock synchronization. O2 looks for a service named `_cs` and when available sends messages to `/_cs/ping` with a reply-to address and sequence number. The service sends the current time and sequence number to the reply-to address. The client then estimates the server’s time as the reported time plus half the round-trip time. All times are IEEE standard double-precision floats in units of seconds since the start of the clock service. O2 does not require or provide absolute date and time values.

This basic synchronization protocol suffers when messages are delayed, so O2 retains information from the last 5 pings and uses the one with the lowest round trip time. Another problem, especially in music systems, is that when clocks are adjusted, carefully timed music sequences can literally skip a beat. In O2, when the local clock is not synchronized, it is sped up or slowed down by 10% until it matches the estimated master clock time. While 10% may seem large, it is not a perceptually large change in terms of musical tempo where the just-noticeable difference is 6-8% [21], and the speed-up or slow-down period typically lasts

for only tens of milliseconds. In the case of very large clock adjustments exceeding 1 second, it is considered musically not useful to remain so far out of synchronization, so the local clock is set to the correct time immediately.

5.4. Scaling Issues. Discovery, clock synchronization, and reliable transmission all add network traffic to an O2 ensemble. O2 is designed to support up to 100 processes in one ensemble. We assume that the ensemble itself does not exceed the network capacity, so our only design constraint is that the *overhead* of using O2 does not overly tax the network or processing time as the number of processes scales up to 100.

One source of sustained network traffic is the clock synchronization protocol, in which each client periodically sends a round-trip UDP packet to the master clock process. We estimate clock time based on the fastest round-trip time to the master in 5 tries. With a polling period of 10 s, the clock will be updated within 50 s. Assuming clock rate differences of 40 PPM (based on standard ± 20 PPM oscillators (<https://www.ctscorp.com/wp-content/uploads/CTS-Corporation-Clock-Oscillator-Timing-Frequency-Electronic-Component-Manufacturer.pdf>)), the worst-case drift in 50 s is 2 ms, which is low in perceptual terms [22]. With a send and reply message every 10 s, 99 clients will generate only 19.8 messages per second. This low polling rate has the disadvantage that processes would take about 40 s to establish synchronization. Instead, the protocol sends the first 15 messages more rapidly, achieving initial synchronization typically within 0.5 s without increasing the long-term network traffic.

Discovery messages are sent periodically and are necessary since a new process could join the ensemble at any time. There is a trade-off between the mean time to join and the density of network traffic. O2 uses a period of 4 seconds, resulting in 25 messages per second in a 100-process ensemble. As with clock synchronization, O2 uses a “fast start” with a 0.133 s initial period between messages. In the worst case, a discovery message is broadcast to all 16 discovery ports in the first 4 seconds, but typically, in a distributed system, O2 processes will use the first discovery port and discovery will take place almost immediately. If the `o2_hub` function is used by a process, discovery messages are not needed and none are sent by the process.

A final source of network overhead is the TCP connections that are maintained between each pair of processes. With 100 processes, there will be 4950 connections, but of course these will be distributed across processes, with each process making 99 connections. These connections are created as part of the discovery process, so when a process joins an ensemble of 99 other processes, there will be a burst of network traffic (about 297 packets) to establish 99 TCP connections. Aside from this somewhat bursty setup behavior, the TCP overhead is limited to acknowledgement (ACK) messages, which only grow in proportion to the number of application-level messages.

5.5. Replies and Queries. O2 has no built-in query system, and, normally, O2 messages do not need replies. Queries have been proposed for OSC but never became widely used, indicating this is not a critical feature. Unlike classic remote

procedure call systems implementing synchronous calls with return values, real-time music systems are generally designed around asynchronous messages to avoid blocking to wait for a reply.

Rather than build in an elaborate query/reply mechanism, we advocate a very simple application-level approach where the “query” sends a *reply-to* address string. The handler for a query sends the reply as an ordinary message to a node under the *reply-to* address. For example, if the *reply-to* address in a `/synth/cpload/get` message is `/control/synthload`, then the handler for `/synth/cpload/get` sends the time back to (by convention) `/control/synthload/get-reply`. Optionally, an error response could be sent to `/control/synthload/get-error`, and other reply addresses or protocols can be easily constructed at the application level.

Although O2’s discovery protocols reveal all the active services in an ensemble, there is no facility to query the namespace of each service or find the parameter types or documentation. However, a directory service was implemented as an O2 service, allowing any other service to register addresses and descriptions [23].

5.6. Address Pattern Matching and Message Delivery. An option in both OSC and O2 is the use of “wildcards” and patterns in addresses, allowing a single message to control multiple parameters. For example, the address `/synth/chan*/alloff` can be used to send a message to `/synth/chan01/alloff`, `/synth/chan02/alloff`, ..., `/synth/chan16/alloff`, assuming these addresses all exist. OSC has been criticized for the need to perform potentially expensive parsing and pattern matching to deliver messages. O2 adds a small extension for efficiency: The client can use the form `!synth/filter/cutoff`, where the initial “!” means the address has no “wildcards.” If the “!” is present, the receiver can treat the entire remainder of the address, “`synth/filter/cutoff`” as a key and do a hash-table lookup of the handler in a single step. This is merely an option, as a node-by-node pattern match of “`synth/filter/cutoff`” should return the same handler function.

6. Performance

O2 is implemented in the C programming language for portability. Performance measurements show that CPU time is dominated by network send and receive time, even when messages are sent to another process on the same host (no network link is involved). Table 2 summarizes measurements where two processes send a message back and forth 2 million times. The fastest time is with O2 and TCP. Perhaps TCP slightly outperforms UDP in these tests because a stateful connection can cache routing or other information. The OSC over TCP performance was about half that of O2. This is likely due to the fact that OSC connections are one-way, and thus *two* TCP connections were opened to send messages back and forth. This prevents acknowledgement (ACK) signals from “piggy-backing” on data packets, doubling the total number of packets. These measurements were run on a 3 GHz Intel Core i7 processor, running OS X v.10.13.6. The main

TABLE 2: Small message send time (just the destination address and a 32-bit integer) for O2 versus OSC and TCP versus UDP. The same communication was also implemented directly in TCP and UDP without any additional layers, sending only one 32-bit integer per message. Run times are wall time, with all messages between two processes on the same host. Averages from multiple runs are reported. Individual runs vary by $\pm 1.5\%$.

	UDP		TCP	
	Time/Message	Messages/Second	Time/Message	Messages/Second
OSC	29 μ s	35,000 s ⁻¹	56 μ s	18,000 s ⁻¹
O2	30 μ s	34,000 s ⁻¹	28 μ s	36,000 s ⁻¹
Direct	22 μ s	46,000 s ⁻¹	20 μ s	44,000 s ⁻¹

conclusion is that O2 features have a negligible impact on performance relative to OSC.

Clock synchronization is difficult to measure. Any technique that can accurately compare clocks on remote machines can be used to synchronize them! However, we can get a good idea of how well clocks are synchronized by observing *estimated* clock differences that are produced by the clock synchronization protocol itself. For example, if the protocol estimates the clock is behind by 3 ms, then the actual clock error is probably 3 ms or less (e.g., it could have been behind by 1.5 ms and is now set to be ahead by 1.5 ms). We ran O2 for 2 hours (740 clock sync periods) on two personal computers sharing a wireless hub/modem that was also being used for Internet access. The median round-trip time was 5.5 ms, but there were 94 round trip times in excess of 100 ms. Nevertheless, the maximum absolute *estimated* clock difference was only 21 ms. The median correction was 0 ms (times were recorded in whole milliseconds). Considering that the just-noticeable difference for rhythmic timing is about 10 ms [22], we conclude that the clock synchronization performance is adequate for music applications, but the algorithm could probably be improved by detecting outliers in the round-trip time.

7. Interoperation

OSC is widely used by existing software. OSC-based software can be integrated with O2 with minimal effort, providing a migration path from OSC to O2. O2 also offers the possibility of connecting over protocols such as Bluetooth (<http://www.bluetooth.org>), MIDI [1], or ZigBee (<http://www.zigbee.org>), although each of these requires extensions to be implemented within the O2 library. Finally, it is possible to create servers to bridge between O2 and other protocols, as illustrated by a WebSockets bridge server.

7.1. Receiving from OSC. To receive incoming OSC messages, call

```
o2_create_osc_port(service, port);
```

which tells O2 to begin receiving OSC messages on *port*, directing them to *service*, which is normally local but could also be remote. Since O2 uses OSC-compatible types and parameter representations, this adds very little overhead to the implementation. If bundles are present, the OSC NTP-style timestamps are converted into O2 timestamps before messages are handed off.

7.2. Sending to OSC. To forward messages to an OSC server, call

```
o2_osc_delegate(service, ip_address,
port, tcp_flag);
```

which tells O2 to create a virtual service (name given by the *service* parameter) to convert incoming O2 messages into OSC messages and forward them to the given *ip_address* and *port*, using a TCP connection if *tcp_flag* is set. Now, any O2 client on the network can discover and send messages to the OSC server.

7.3. Other Transports. Handling O2 messages from other communication technologies poses two interesting problems: What to do about discovery, and what exactly is the protocol? Our goal is to allow the O2 API to be supported directly on clients and servers connected by non-IP technologies. We do this by having an O2 process forward messages to and from non-IP hosts. As an example, let us assume we want to use O2 on a Bluetooth device (we will call it Process D; see Figure 2) that offers the `Sensor` service. We require a direct Bluetooth connection to Process B running O2. Process B will claim to offer the `Sensor` service and transmit that through the discovery protocol to all other O2 processes connected via TCP/IP. Any message to `Sensor` will be delivered via IP to Process B, which will then forward the message to Host D via Bluetooth. Similarly, programs running on Host D can send O2 messages to Process B via Bluetooth where the messages will either be delivered locally or be forwarded via TCP/IP to their final service destination. It is even possible for the destination to include a final forwarding step though another Bluetooth connection to another computer; for example, there could be services running on computers attached to Process C in Figure 2. The same approach is used for other transports such as ZigBee or serial links such as RS-232.

In addition to addressing services, O2 sometimes needs to address the O2 subsystem itself; e.g., clock synchronization runs even in processes with no services. Services starting with digits, e.g., “128.2.60.110:8000,” are interpreted as an IP:Port pair. To reach an attached non-IP host, a suffix may be attached; e.g., Host D in Figure 2 can be addressed by “128.2.60.110:8000:bt1.”

“Other transports” are not limited to networks. Recent work has explored the use of shared-memory lock-free queues to send O2 messages to high-priority threads in the same process, providing synchronized communication without locks. This is particularly useful in real-time music audio

O2 Spy

Application: IP for O2 Hub (Optional)

Service:

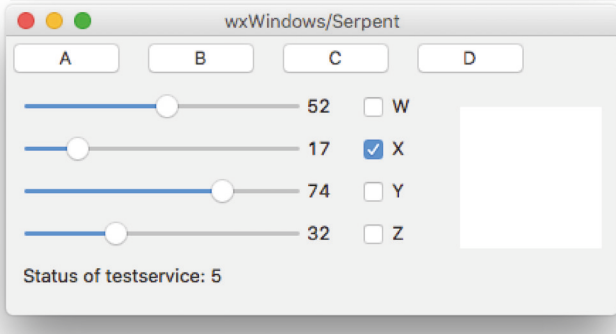
Service Status: **Remote**

Start O2
 Install Tap
 Pause Output
 Show O2 Status Messages

```

o2_message: /o2spy/canvas@0 "fff" (2,88,0)
o2_message: /o2spy/canvas@0 "fff" (4,91,0)
o2_message: /o2spy/canvas@0 "fff" (8,94,0)
o2_message: /o2spy/canvas@0 "fff" (10,97,0)
o2_message: /o2spy/canvas@0 "fff" (12,100,0)
o2_message: /o2spy/sliderD@0 "f" (33)
o2_message: /o2spy/sliderD@0 "f" (32)
o2_message: /o2spy/sliderD@0 "f" (31)
o2_message: /o2spy/sliderD@0 "f" (32)
o2_message: /o2spy/buttonB@0 "f" (1)
o2_message: /o2spy/sliderB@0 "f" (25)
o2_message: /o2spy/sliderA@0 "f" (43)
o2_message: /o2spy/sliderC@0 "f" (74)
o2_message: /o2spy/checkX@0 "f" (1)
o2_message: /o2spy/checkZ@0 "f" (1)
o2_message: /o2spy/checkZ@0 "f" (0)
o2_message: /o2spy/sliderA@0 "f" (52)
o2_message: /o2spy/checkW@0 "f" (1)
o2_message: /o2spy/sliderB@0 "f" (17)
o2_message: /o2spy/checkW@0 "f" (0)

```



wxWindows/Serpent

A B C D

52 17 74 32

W X Y Z

Status of testservice: 5

FIGURE 4: Web pages can access O2 via a WebSockets interface to a local server. Here, a web page is used to tap into messages delivered from a graphical control panel process (at bottom) to a remote receiver process (not shown).

applications where locks are typically forbidden within audio computation threads or callback functions. Applications will see audio computation as an O2 service that can be addressed in the normal way. Messages will be delivered by TCP/IP to the right process, and, from there, messages can be forwarded to the high-priority audio service by appending them to a lock-free queue.

7.4. Language Support. O2 is currently implemented in the C programming language for portability and to simplify linking with programs written in other languages. Serpent [24], a real-time scripting language developed especially for interactive music applications, includes O2 in the standard release. O2 has also been incorporated into Kronos [25] and used to create a network-based audio synthesis server. In this system, real-time audio is streamed via O2 messages [23].

7.5. WebSocket Support. An interesting recent development is a server that enables O2 access from web pages using WebSocket technology. The server, written in Serpent, is a lightweight HTTP server with WebSocket capability. The

server is normally run on the local host along with a web browser. Any page loaded into the browser can open `ws://localhost:8080/` to create a WebSocket connection to the local server that also runs O2. A simple protocol is implemented over a WebSocket to enable the web page to join an O2 ensemble, create services, and send and receive O2 messages, including OSC messages. The use of WebSockets adds an extra hop to message delivery but avoids the security and practical problems of writing extensions for a variety of web browsers.

Figure 4 illustrates a web-based application that can monitor remote O2 message delivery using the `o2.tap` function described earlier. The O2 monitor application, implemented in JavaScript and HTML, connects over a WebSocket to the local server that is running as an O2 process.

The WebSocket interface to O2 also creates the possibility for applications written in other languages such as Python, Java, C#, or Ruby to access O2 through existing WebSocket libraries rather than creating a “foreign function interface” library for O2 in each language.

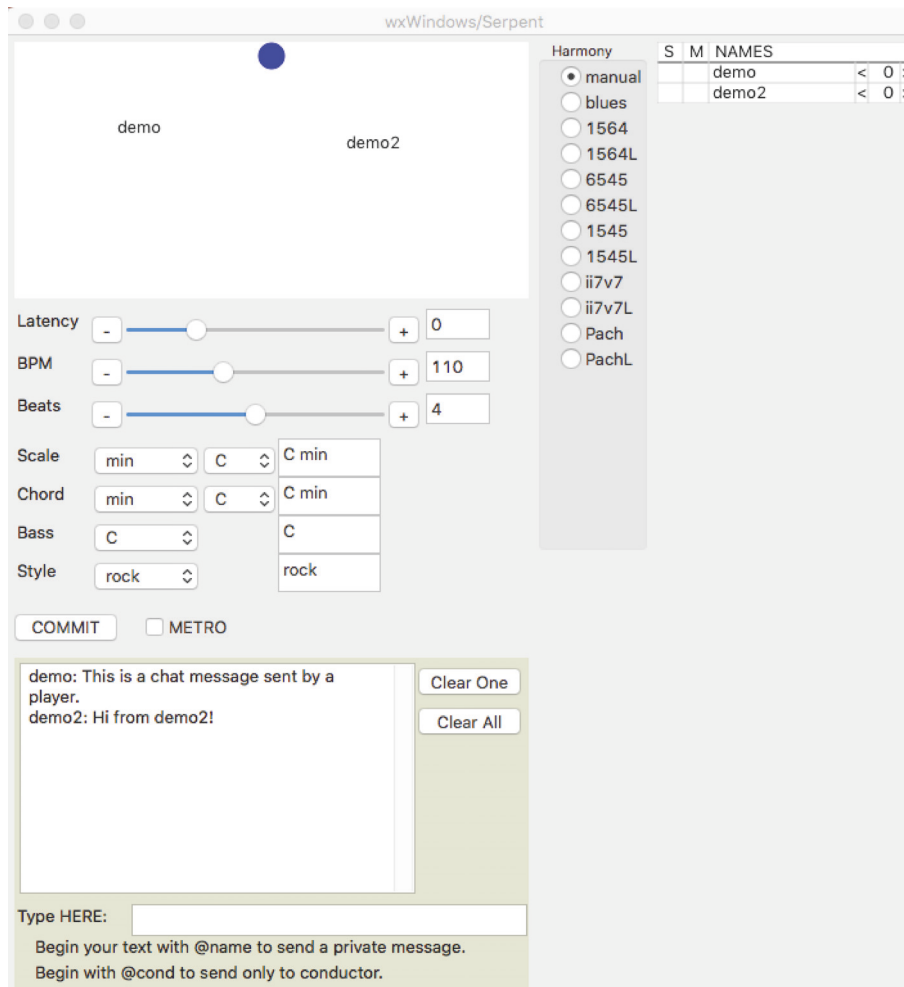


FIGURE 5: Control panel for the Conductor of the CMU Laptop Orchestra. The top left window is a map of the stage showing the locations of all the players (only 2 are shown here). The circle moves to indicate the beat and tempo. At middle left are controls for tempo, meter, harmony, and style, which are sent to all players when the COMMIT button is pressed. At the bottom is a chat window allowing players to communicate during the performance. The conductor also has a number of standard chord progressions that can be selected below “Harmony” (top, center). When players join the ensemble, their names are displayed in a list. To the left of each name is a solo (S) and mute (M) button; to the right are softer (<) and louder (>) buttons surrounding the current loudness offset (0 in this example).

8. Example

One substantial example of O2 in practice has been recent performances by the CMU Laptop Orchestra. Originally built around TCP/IP and OSC, this networked performance uses a central *conductor* to send tempo, key, meter, and style information to around 20 to 25 client computers (see Figure 5). Each client is a semiautonomous *player* that follows the musical structure and constraints of the conductor but also has real-time controls operated by a human (who is also the creator/programmer of the player). Players fill different musical roles such as bass, melody, arpeggiator, chordal accompanist, or drummer, and controls include mobile devices running TouchOSC (<https://hexler.net/software/touchosc>) and connected over Open Sound Control. A human “semiconductor” can change tempo, meter, and key and also mute, unmute, and adjust the volume of individual players.

Players initially send their names to the conductor service, which keeps a list of active players by checking their status periodically. To obtain musical synchronization, the conductor expresses beat times as a linear function: the time for beat b , $f(b) = a + b/s$, where a is the (theoretical) time of beat zero, and s is the tempo in beats per second. Only a and s need to be delivered to clients when the tempo changes, and the delivery time is not critical since a and s are not time dependent. All players compute the same value $f(b)$ for each beat, and all players have synchronized clocks, so beats are accurately synchronized. In fact, the main impediment to audio synchronization is variability in the latency of various software synthesizers and audio device drivers. Each player schedules output ahead of time according to an audio-latency compensation parameter that users can adjust, resulting in synchronization to within a few milliseconds. Readers can view performances online at <https://youtu.be/icLUJMM-11M>

and <https://youtu.be/L-Sar4D7IYY>. These performances used WiFi to simplify the setup.

9. Summary and Conclusions

O2 is a new protocol for real-time interactive music systems. It can be seen as an extension of Open Sound Control, keeping the proven features and adding solutions to some common problems encountered in OSC systems. In particular, O2 allows applications to address services by name, eliminating the need to manually enter IP addresses and port numbers to form connected components. O2 offers two classes of messages so that “commands” can be delivered reliably, and sensor data can be delivered with minimal latency. In addition, O2 offers a standard clock synchronization and time-stamping system that is suitable for local area networks. We have implemented O2 and shown that its speed is comparable to an Open Sound Control implementation. Although O2 assumes that processes are connected using TCP/IP, we have also described how O2 can be extended over a single hop to computers via Bluetooth, ZigBee, RS-232, or other communication links, and how a WebSockets-to-O2 bridge server can open O2 applications to web browsers.

A number of extensions are possible, and future work includes extensions for audio and video streaming, and dealing with network address translation (NAT). We are also working on “externals” for Pd [26] and Max/MSP [27], which are widely used development platforms in the computer music community. As Zeroconf (Bonjour) becomes standard, we believe we can abandon our self-contained discovery system in favor of a standard one. O2 has been run in networks of 25 hosts, but it would be interesting to measure performance on larger networks, at least up to 100 hosts. Overall, we believe O2 is a good candidate for OSC-like applications and a variety of networked mobile and IoT devices in the future.

Data Availability

The source code for O2 is available for commercial and noncommercial use at <https://github.com/rbdannenberg/o2>. The source code and executable versions of the Serpent programming language are available for commercial and non-commercial use at <https://sourceforge.net/projects/serpent/>.

Disclosure

This paper is an extensively revised and extended version of an earlier conference publication.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

Thanks are due to Adrian Freed for comments on an earlier paper. Zhang Chi contributed to the initial implementation of

O2 and Hongbo Fang implemented a WebSockets protocol in Serpent. O2 has developed and evolved through many interactions with students, visitors, and faculty in the School of Computer Science at Carnegie Mellon University.

References

- [1] J. Rothstein, *MIDI: A Comprehensive Introduction*, A-R Editions, 2nd edition, 1995.
- [2] M. Wright, A. Freed, and A. Momeni, “OpenSound control: state of the art 2003,” in *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03)*, pp. 153–159, Montreal, Canada, 2003.
- [3] M. Wright, A. Freed, A. Lee, T. Madden, and A. Momeni, “Managing complexity with explicit mapping of gestures to sound control with OSC,” in *Proceedings of the International Computer Music Conference*, pp. 314–317, International Computer Music Association, Habana, Cuba, 2001.
- [4] E. Lynch and J. Paradiso, “Sensorchimes: musical mapping for sensor networks,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 137–142, Brisbane, Australia, 2016.
- [5] R. Fiebrink, P. R. Cook, and D. Trueman, “Play-along mapping of musical controllers,” in *Proceedings of the 2009 International Computer Music Conference, ICMC*, pp. 61–64, Canada, 2009.
- [6] J. Malloch, S. Sinclair, and M. M. Wanderley, “Distributed tools for interactive design of heterogeneous signal networks,” *Multimedia Tools and Applications*, vol. 74, no. 15, pp. 5683–5707, 2015.
- [7] G. Essl, “Automated ad hoc networking for mobile and hybrid music performance,” in *Proceedings of the International Computer Music Conference 2011*, pp. 399–402, Huddersfield, 2011.
- [8] D. Trueman, P. Cook, S. Smallwood, and G. Wang, “PLORk: the Princeton Laptop Orchestra, year 1,” in *Proceedings of the International Computer Music Conference, ICMC 2006*, pp. 443–450, 2006.
- [9] R. B. Dannenberg, S. Cavaco, and E. Ang, “The Carnegie Mellon Laptop Orchestra,” in *Proceedings of the 2007 International Computer Music Conference*, vol. II, pp. II-340–II-343, The International Computer Music Association, ICMA, San Francisco, USA, 2007.
- [10] G. Hajdu, “Embodiment and disembodiment in networked music performance,” in *Body, Sound and Space in Music and Beyond: Multimodal Explorations*, C. Wöllner, Ed., pp. 257–278, Routledge, Abingdon-on-Thames, 1st edition, 2017.
- [11] R. B. Dannenberg and T. Neuendorffer, “Scaling up live internet performance with the global net orchestra,” in *Proceedings of the 11th Sound & Music Computing Joint with the 40th International Computer Music Conference*, pp. 730–736, Athens, Greece, 2014.
- [12] S. Gresham-Lancaster, “The aesthetics and history of the hub: the effects of changing technology on network computer music,” *Leonardo Music Journal*, vol. 8, pp. 39–44, 1998.
- [13] M. Wright, “Open Sound Control: an enabling technology for musical networking,” *Organised Sound*, vol. 10, no. 03, p. 193, 2005.
- [14] E. Brandt and R. B. Dannenberg, “Time in distributed real-time systems,” in *Proceedings of the International Computer Music Conference*, 1999.
- [15] M. Henning, “The rise and fall of CORBA,” *Queue*, vol. 4, no. 5, pp. 28–34, 2006.

- [16] E. Guttman, "Autoconfiguration for IP networking: Enabling local communication," *IEEE Internet Computing*, vol. 5, no. 3, pp. 81–86, 2001.
- [17] A. Eales and R. Foss, "Service discovery using open sound control," in *Proceedings of the AES 133rd Convention 2012*, AES, pp. 348–354, San Francisco, USA, 2012.
- [18] J. Narveson and D. Trueman, "LANdini: a networking utility for wireless LAN-based laptop ensembles," in *Proceedings of the Sound and Music Computing Conference (SMC)*, Stockholm, Sweden, 2013.
- [19] F. Cristian, "Probabilistic clock synchronization," *Distributed Computing*, vol. 3, no. 3, pp. 146–158, 1989.
- [20] S. Madgwick, T. Mitchell, C. Barreto, and A. Freed, "Simple synchronisation for open sound control," in *Proceedings of the 41st International Computer Music Conference*, pp. 218–225, Denton, TX, USA, 2015.
- [21] K. Thomas, "Just Noticeable Difference and Tempo Change," *Journal of Scientific Psychology*, 2007.
- [22] A. Friberg and J. Sundberg, "Perception of just-noticeable time displacement of a tone presented in a metrical sequence at different tempos," *STL-QPSR*, vol. 34, no. 2-3, pp. 49–56, 1993.
- [23] V. Norilo and R. B. Dannenberg, "KO2 distributed music systems with O2 and Kronos," in *Proceedings of the 15th Sound and Music Computing Conference (SMC2018)*, 2018.
- [24] R. B. Dannenberg, "A language for interactive audio applications," in *Proceedings of the 2002 International Computer Music Conference*, pp. 509–515, International Computer Music Association, San Francisco, USA, 2002.
- [25] V. Norilo, "Kronos: a declarative metaprogramming language for digital signal processing," *Computer Music Journal*, vol. 39, no. 4, pp. 30–48, 2015.
- [26] M. Puckett, "Pure data," in *Proceedings of the International Computer Music Conference*, pp. 224–227, International Computer Music Association, San Francisco, CA, USA, 1996.
- [27] M. Puckette, "Max at Seventeen," *Computer Music Journal*, vol. 26, no. 4, pp. 31–43, 2002.

Research Article

The Influence of Coauthorship in the Interpretation of Multimodal Interfaces

Fabio Morreale ¹, Raul Masu,² and Antonella De Angeli³

¹*Creative Arts and Industries, University of Auckland, New Zealand*

²*Madeira-ITI and FCT/Universidade Nova de Lisboa, Portugal*

³*Faculty of Computer Science, Free University of Bozen-Bolzano, Italy*

Correspondence should be addressed to Fabio Morreale; f.morreale@auckland.ac.nz

Received 15 January 2019; Accepted 2 April 2019; Published 24 April 2019

Guest Editor: Federico Avanzini

Copyright © 2019 Fabio Morreale et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a model to codesign interpretively flexible artefacts. We present the case study of Beatfield, a multimodal system that allows users to control audiovisual material by means of tangible interaction. The design of the system was coauthored by individuals with different background and interests to encourage a range of difference interpretations. The capability of Beatfield to foster multiple interpretations was evaluated in a qualitative study with 21 participants. Elaborating on the outcome of this study, we present a new design model that can be used to stimulate heterogeneous interpretations of interactive artefacts.

1. Introduction

Throughout history, many musical instruments gained an ample uptake for having been used in ways that designers did not originally envision [1]. This is the case, for instance, of the tape recorder, which was transformed into a compositional tool by composers and engineers in the 19th century [2], and of the turntable, which found a new identity in the hands of DJs [1]. In more recent years, several musical interfaces have been appropriated by players in ways that designers did not envision [1, 3]. The very nature of interactive devices and musical instruments, it seems, becomes clear only when submitted to the verdict of the final users, whose possibility for appropriating and repurposing technology gave rise to the hacking culture at the end of last century.

Only in recent years did researchers in human-computer interaction (HCI) start focusing their attention on the phenomena of interface appropriation [4, 5], hacking, and making communities [6–8]. Prior to that, the objective of HCI designers was to convey a single, clear interpretation of their design artefacts. This objective was a direct consequence of the contexts in which HCI operated at that time, i.e., the work environment, which required research on human factors to achieve better ergonomics and increased productivity [9].

However, due to its recent intersection with new domains, in particular the arts and humanities [9], the HCI community started questioning this approach, suggesting that multiple interpretations of an interactive system can coexist [10, 11]. A considerable corpus of work was then produced advocating for the importance of *appropriating* and even *subverting* interactive systems [11–13]. Researchers and designers started exploring ways to design interactive systems that can be easily appropriated by the user. To date, however, no work has pinpointed a series of design activities that allow a single interactive artefact to present itself in different ways to different user and thus stimulate a variety of interpretations.

In this paper, we discuss our attempt to address this challenge. We introduce the idea that the design of interpretively flexible systems should embed multiple values and backgrounds at design stage. To this end, we propose a codesign method based on a number of activities in which a few participants coauthor a new interactive system. These activities, inspired from a design framework of musical interfaces [14], were tested to guide the development of Beatfield, an interactive music system designed to be interpretively flexible (Figure 1). We empirically evaluated the extent to which Beatfield managed to stimulate a variety of interpretations in a field study with 21 participants. A qualitative analysis

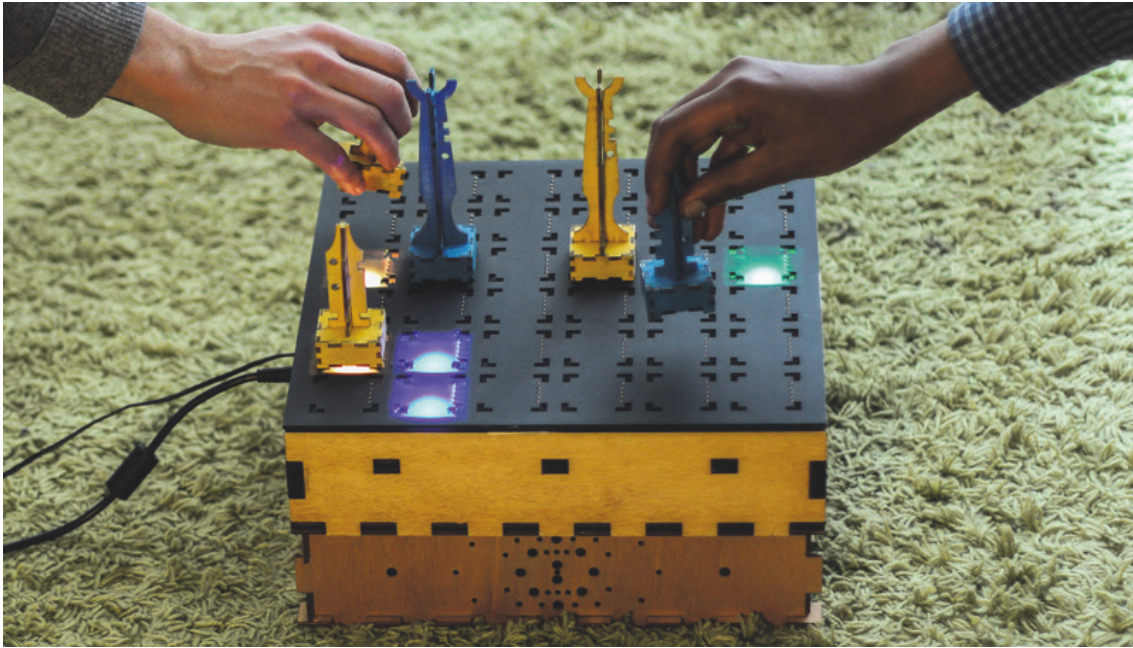


FIGURE 1: Beatfield, an interactive music system developed as a research through design artefact.

was performed on interview transcripts and video analysis. The results confirmed that Beatfield successfully elicited a variety of interpretations about its own nature and the way to interact with it. Elaborating on these results, we propose discussions that offer new insights on human interaction with multimodal devices.

The remainder of this paper is structured as follows. Section 2 reviews the related work; Section 3 describes the design process that led to the development of Beatfield, whose implementation is presented in Section 4. The field study is presented in Section 5, and its results are discussed in Section 6. The paper concludes with reflections on the design of open-meaning multimodal systems.

2. Related Work

2.1. Design for Open Interpretation. The idea of an “interpretively flexible” artefact first appeared in [11], who described it as a system in which *meaning is coconstructed by users and designers*. The resulting system is described as a sort of “Rorschach interface” onto which each user would project their own personal meanings. A number of design suggestions were also identified to encourage users to appropriate and reinterpret a system and produce their own meanings [11]. For instance, the system should not constrain the user to a single interaction mode; instead, it should provide information about the topic without specifying how to relate to it and stimulate original interpretations while discouraging expected ones. Similarly, Gaver and colleagues suggested that if the ultimate purpose is to foster the association of personal meanings with a design artefact, any clear narrative of use should be avoided [13]. Multiple interpretations of a piece of design can be promoted by embedding in the

design ambiguous situations that require users to participate in meaning-making [15].

Even though *ambiguity* had been avoided in HCI for a long time, researchers are now advocating that it can be considered as a resource for design [15–17]. Three kinds of ambiguity in design can be distinguished [15]: *ambiguity of information*, which arises when the information is presented in an ambiguous way; *ambiguity of context*, which occurs when things assume different meanings according to the context; and *ambiguity of relationship*, which is related to the user’s personal relationship with a piece of design. A series of design considerations were also proposed to foster an ambiguous reception. For instance, the product should expose inconsistencies and cast doubts on the source of information; it should assemble disparate contexts, as to create tensions that must be resolved; and should diverge from its original meaning when used in radically new contexts.

HCI researchers also proposed to support open interpretations of artefacts by embedding elements of *appropriation*, i.e., “improvisations and adaptations around technology”[5]. A number of suggestions to design for appropriation have been suggested by Dourish [4] and Dix [5], among which (i) including elements where users can add their own meanings [5]; (ii) offering tools to accomplish a task without forcing an interaction strategy [4, 5]; and (iii) supporting multiple perspectives on information [4]. Another factor that can encourage multiple interpretations of a design artefact is *randomness* [18]. Randomness provides users with positive and rich experiences as, arguably, it is the very experience of unpredictability that mostly captures user imagination. Our proposed codesign model uses as design material some of the suggestions for designing for open interpretations in the context of interactive systems.

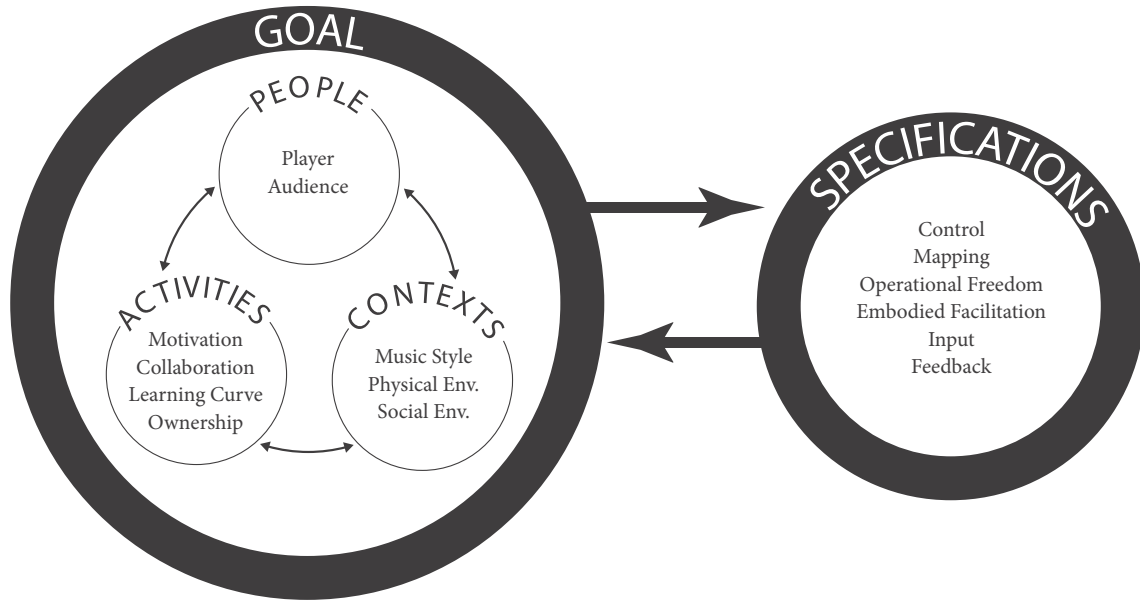


FIGURE 2: MINUET, a design framework of musical interfaces [14].

2.2. Interactive Music Systems. The design space of interactive music systems unfolds along several dimensions, which have been categorised by several researchers in a number of design frameworks [14, 19, 20]. This design space can be roughly described along a continuum that stretches from digital musical instruments (DMIs) to interactive installations [19]. DMIs are innovative devices to perform music: they can either take a form that resembles that of existing instruments [21, 22] or be completely new interfaces [23, 24]. At the other end of the spectrum, there are interactive systems, which are designed with a focus on the experience of the player rather than the actual musical output [14]. This is the case, for instance, of The Music Room, which was ideated to stimulate collaborative experiences among players [25]. In other cases the objective seems to purposely stimulate ambiguous reception. Polyfauna, for instance, is a mobile app that allows the user to influence audio and visual aspects, while the mapping between the input and the output remains ambiguous. As another example of enigmatic mapping, Cembalo Scrivano is an augmented typewriter that responds to user interaction with ambiguous sounds and visuals [26].

In parallel to the development of novel music creation tools, new relations between different actors (i.e., designers, composers, and performers) emerged [27]. Schnell and Battier [28] proposed the concept of the *composed instrument*, an interactive music system that is both an instrument and a score. Under this perspective, the design of an interactive music system coincides with the compositional process and the authorship of the resulting artistic outcome is embedded in the technology itself. In these cases, the composer is usually also the designer and imposes her own aesthetic on the artefact. Recently, a codesign activity has been proposed to collaboratively develop musical interfaces [29].

The codesign model that we present in this paper is inspired by MINUET, a framework aimed at assisting the design of interactive music systems centred on the player experience [14]. MINUET is structured as a design process consisting of two stages, goal and specifications (Figure 2). At the first stage, *Goal* frames a conceptual model of the reason for existence of the interface by considering three separate *entities*: people, activities, and contexts. *People* addresses the designer's objectives from the point of view of the target category of players and from the role of the audience; *Activities* addresses the character of the envisioned interaction; *Contexts* addresses the environment and the physical setup of the interface. Each of these *entities* is composed of a number of *concepts* that consider design issues on a more pragmatic level (e.g., *activities* are defined by the concepts *motivation*, *collaboration*, *learning curve*, and *ownership*). The second stage of the design process, *specifications*, delves into the goals from the point of view of the interaction constraints between the player and the artefact. Concepts considered at this stage are control, mapping, input, feedback, operational freedom, and embodied facilitation. *Control* refers to the extent to which the player has control on the music output. *Mapping* refers to the relation between the user input and the musical output. It can be convergent (a sequence of actions produces a single sound), or divergent (a single action affects many musical factors). *Input* concerns the modality of interaction (e.g., visual, tactile, semantic). *Feedback* considers different feedback modalities beyond sound. *Operational freedom* indicates the extent to which the instrument can push the player to have a creative, flexible interaction. *Embodied facilitation* invites to consider whether or not the design of the interface should impose limitations by suggesting specific interaction modalities. In the work presented in this paper, we modelled the proposed design activities on MINUET to

foster reflections on both abstract and practical level when designing an interface that is interpretively opened.

3. Beatfield: Design

The goal of this project was to generate design activities aimed at creating an interpretively flexible artefact, as defined in Section 2.1. As opposed to the suggestion offered by [11], who endorsed the centrality of the designer in the process when designing interpretively flexible systems, we propose the idea that the designer should in fact resist embedding his own aesthetics in the design of the system. Rather, we advocate for several individuals to coauthor the system through codesign sessions, thus avoiding privileging any particular interpretation.

We tested this model in an empirical study with a multidisciplinary team composed of a researcher in HCI, a musician, a game designer, and a visual artist. This team was involved to propose a number of possible scenarios. MINUET [14] appeared to be the ideal framework to guide this process as it was specifically designed to elaborate ideas and objectives when designing an interactive music system. However, MINUET does not provide practical guidance on how to put these ideas into practical design activities. To this end, we created a design activity divided into two parts as to mirror the structure of MINUET. Specifically, the goal of Part I is to outline a new interactive music system; and Part II is dedicated to embed in its design elements that make it open to interpretation.

3.1. Part I: Coauthoring a New Interactive Music System. The first step of MINUET is intended to generate a conceptual model of the interaction, i.e., the user story at very high level. We operationalised this step in the following way. The researcher prepared a block of papers to be handed to each participant. Each block was composed of three sheets, each describing one of the *entities* specified in MINUET and its associated *concepts*. Being each entity equally important in the design of an interactive music system [14], the order of each entity was shuffled across the three papers to avoid privileging any of them. As an example, the order of the entities and their associated concepts in one of the blocks was sheet 1 - Context; sheet 2 - Activities; and sheet 3 - People. The design activity took place in a “brainstorming” room that contained several facilities and appliances to stimulate design activities. The researcher moderated the sessions. To start with, he explained that the goal of the activity was to conceptualise a new interactive music system. The research objective to create an open-interpretation artefact was purposely not disclosed at this point.

For the first task, each participant came up with a scenario using the *entity* on sheet 1 and the associated *concepts*. Participants were invited to draw or write down their ideas. After 10 minutes, the blocks were collected and redistributed in a different order among the participants, who were asked to read the scenario described in the first sheet and to elaborate it further, this time using the entity indicated at sheet 2. The same process was repeated a third time: participants had to finalise the scenario authored by the other two participants

using the *entity* of the last sheet. At the end, all scenarios were coauthored by all participants, and all participants contributed to a different *entity* in each scenario. Finally, the three scenarios were presented and discussed and the researcher then invited the team to come up with a single scenario. The outcome of this activity was the conceptual proposition of a tangible multimodal interface that we eventually named Beatfield. The high-level user story of Beatfield as defined at the end of the activity follows: *Beatfield is an ambiguous tangible exploration of an audiovisual landscape. A musical source is placed inside a box; players can let the music spawn from it by placing objects on top of the box. Each time a new object is positioned on the board, a music pattern would play.*

3.2. Part II: Making the System Open to Interpretation. The second part of the workshop consisted of defining the specifications of the interface in a way to stimulate multiple interpretations. Only at this point did the researcher disclose the actual objective of the project to build an interface that was interpretively flexible. The activity took the form of a focus group. At the beginning of the session, the researcher handed each participant a paper containing a list of design suggestions as identified from the related work on open interpretation of design artefacts (Table 1). Participants were invited to reflect upon the user story defined in the first part of the workshop and to detail possible design specifications using the *concepts* associated with this stage of MINUET. Each participant was then given 30 minutes to individually elaborate upon each concept adopting the design suggestions described in the list. At the end, each *concept* was discussed in group. The descriptions of each *concept* were iteratively refined several times before profiling the final versions, which is detailed below using *concepts* from MINUET.

3.2.1. Input. Players can control music and visuals by positioning pieces on a game board. A total of eight pieces are available, equally divided into two colours (blue and yellow). Each colour group is composed of one king and three pawns, which have different roles. The craft of the pieces was later commissioned to an artist that was asked to (i) take inspiration from chess pieces while at the same time departing from them (using chess pieces in a different context might stir ambiguous receptions among players [15]), (ii) suggest that the two pieces have different roles, and (iii) evoke an enigmatic imagery. The initial sketches and the final pieces are shown in Figure 3.

3.2.2. Feedback. Given the multimodal character of the system, we opted for auditory, tactile, and visual feedback. Priority was given to sound given its lower semantic character compared to visuals, which might lead to higher variability of interpretations. Some simple visual feedback would though be included to add a new layer of interpretation.

3.2.3. Control. Players can interact with some aspect of the music at a high level. Specifically, they are allowed to control the rhythm of the composition, leaving harmony and melody selection to the computer. The *mechanics* of the interaction would be defined: placing an object on the

TABLE 1: List of design suggestions that we elaborated from related work. These suggestions were offered to the codesign participants as design material to stimulate idea generation and constitute the boundaries of our intervention in the design process.

Design suggestions	Source
Stimulate new interpretations by purposefully blocking expected ones.	[11]
Offer unaccustomed roles to encourage imagination.	[15]
Do not make everything in the system have a fixed meaning, but include elements where users can add their own meanings.	[5]
Use randomness as a creative tool used to generate interesting content leaving the users interpret the output.	[18]
Gradually unfold new opportunities for interpretation over the course of interaction.	[11]
Frustrate any consistent interpretation.	[11]
Bringing together disparate contexts to create a tension that must be resolved.	[15]
Set false expectations by demonstrating intended interactions and resetting expectations.	[36]
Offering unaccustomed roles to encourage imagination.	[15]

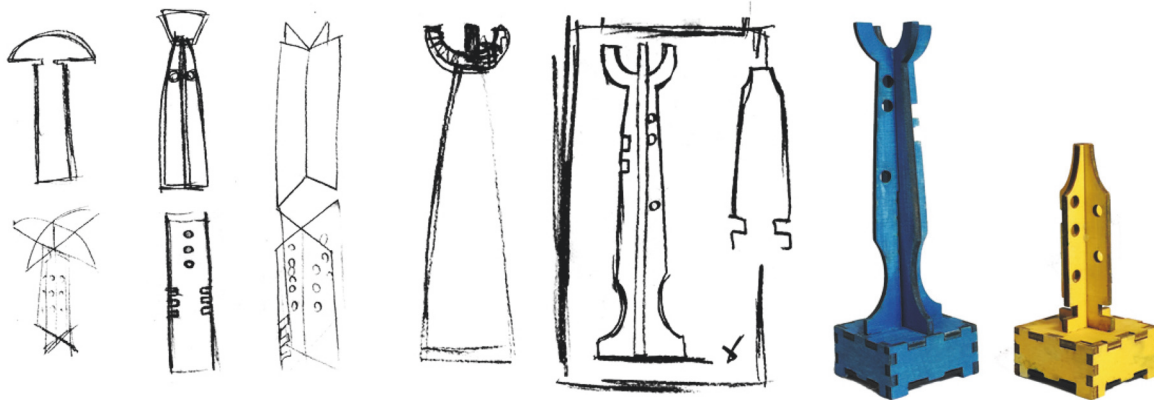


FIGURE 3: Initial sketches of the pieces and final builds.

game board always results in the same outcome. However, the rules and roles of the interaction would be withheld to the players, an approach that contrasts traditional games, in which players have a specific role that is made for them by the game, “a set of expectations within the game to exercise its effect” [30]. Rather, we denied the players to build expectations, thus allowing them to play accordingly to their own wishes and decide their own strategies. Players can build formal definitions of their interaction (e.g., they agree on a turn-based system), thus adopting a “voluntary act of unnecessary overcome obstacles” [31], and become actual players. For instance, players might or might not give pieces of different colours the identity of teams; also, there is no specific need for cooperative or competitive behaviours, but both behaviours have the potential to emerge. It is a design without expectations, which allows the experience to exceed the boundaries of belonging to a particular domain, being it game, music, or artwork. This idea resonates with the concept of the well-played game [32]: a well-played game cannot be found in a specific game, but rather in the experience, in the spirit of playing itself. Games can be “well-played” as long as they are continuously created and players are encouraged to change their standard rules.

3.2.4. Mapping. The mapping between user interaction and generated music is divergent and nontransparent. Player actions are not directly associated with any specific sound.

Rather, an algorithmic agent determines the evolution of the piece; thus, the player fails to precisely recognise his gesture in a direct musical output. The mapping between the physical objects and the music is the result of a design process meant to stimulate reflections in the player and to remain ambiguous.

3.2.5. Operational Freedom. Players are free to guess the functionality of the system and to decide their own objective of interaction based on their interpretation of the system. To this end, any information and narrative of use about the functionalities and the objectives of the interface are withheld from players.

3.2.6. Embodied Facilitation. Beatfield is intended to offer limited freedom of interaction. There is a set of moves and strategies that are considered acceptable and that are necessary for the achievement of an effect in the game (e.g., the player can place a piece between two cells of the board, but cannot expect any outcome). However, the system has not been developed considering the emergence of any particular interaction nor assuming a particular behaviour of the users. In other words, while there cannot be unlimited ways to use the system, it does not exist the right way either.

4. Beatfield: Implementation

We iteratively prototyped the technological apparatus to accommodate the specifications outlined in the previous

section. The final configuration of Beatfield is shown in Figure 1. The interface is composed of an augmented game board that sits on top of a wooden box containing a wireless portable speaker. The chosen speaker outputs sound on two opposite directions; thus, sounds can be reproduced with the same intensity from all the four sides, which avoids Beatfield to have one correct orientation.

4.1. Gesture Sensing. The physical interface was an adaptation of an existing embedded system developed by colleagues at the University of Trento called *Radiant*². The *Radiant*² is a 32 cm² wooden box whose surface (the game board) is divided into 36 5x5 cm cells organised as a 6x6 matrix [33]. Each cell is equipped with electronic contacts that allow communications with the blocks and with magnets that facilitate the blocks to slot in. The top cover is made of a thin sheet of translucent black plastic, beneath which lays a matrix of RGB LEDs, one for each cell. Players can interact with the interface by positioning custom made wooden blocks on the surface. The base of each block embeds a number of microcontrollers, custom electronics boards, and surface contacts that allow to exchange information with the cell of the *Radiant*². On top of the base, we slotted in the crafts of the pieces as shown in Figure 3. To facilitate the mechanical placement, the corners of each cell have L-shaped cuts that are aligned with the bottom corners of the blocks, which have the same L-shape but are extruded. This mechanical guidance is supported by the use of magnets hidden under the surfaces; the magnetic attraction between a cell of the *Radiant*² and the block facilitates the electrical connection. Being the base of each block a square, there are four possible orientations; therefore, the pieces can be detected regardless of their orientation. In order to obtain a fully portable system, an integrated Wi-Fi network was included inside the *Radiant*² to communicate with an external computer.

4.2. Game Mechanics. The mechanics of the game are illustrated in Figure 4.

4.3. Music Generation. The sound design and the mapping between the physical objects and the music are the result of a design process meant to be open to different interpretations. Beatfield generates in real time a tune that is divided into two main components: a *drone tone* and a number of *rhythmic patterns*, whose notes are based on a global harmony. The drone is always active as a background sound, even in idle condition, while the rhythmic patterns only play following user interaction. Specifically, a rhythmic pattern is triggered and looped each time a *pawn* is placed on a cell: the more *pawns* are added to the board, the more rhythmic patterns are generated. When a *pawn* is removed from the board, the corresponding rhythmic pattern is turned off. A global metronome set at 60 BPM synchronises all the rhythmic patterns and the drone tone. The global harmony consists of a set of 8 notes forming an atonal scale to control the overall harmonic coherence of the music. This set of notes cyclically changes every 12 beats; at each new cycle one of the notes is removed from the set of available ones and replaced with

a new one that is randomly chosen. This new note is also transposed two octaves lower and replaces the previous note of the drone. As a result, each 12 beats the drone plays a new note, resulting in a continuously changing background.

4.3.1. Rhythmic Patterns. Each cell of the board is associated with a specific rhythm according to the following mapping. Different columns are associated with different lengths of the bar: the leftmost column is one-quarter bar long, the one on its right is two quarters long, and so on up to six quarters in the rightmost column. The rows determine the number of notes in the bar: the topmost row has one note per bar, the second has two notes, and so on up to six notes for bar in the bottom row (Figure 5). This configuration allows for a variety of rhythmic combinations, whose density spans from the cell positioned in [1, 1], which plays six notes per second, to the cell positioned in [6, 6], which plays one note every six seconds.

4.3.2. Sound Design. Both the drone and the rhythmic patterns are generated using FM synthesis developed in Max/MSP. The modulation index and the carrier-modular ratio are used to differentiate the timber of the patterns according to the colour of the associated *pawn*. Blue *pawns* have bell-like low timber, while yellow *pawns* have bright timber. The modulation index also reacts to the colour of LED lights. When a *pawn* is positioned on top of a cell illuminated with its same colour the harmonic spectrum of the corresponding rhythmic pattern is synthesised with higher harmonics by increasing the modulation index.

5. Field Study

We evaluated the capability of Beatfield to foster flexible interpretations in a field study. We selected 21 participants with different backgrounds and artistic knowledge to foster diverse receptions of the system. Some were invited to try it alone (N=9), while others in group of two (N=6) or three (N=6).

The study took place in an empty room in a historical building in the city centre of Trento, in Italy. The room was lit up by four dim lights and Beatfield was positioned on a table at the centre of the room (Figure 6). A visible camera recorded the players' activities. In between session, the pieces were shuffled on the table next to the *Radiant*², as to prevent specific piece arrangements (e.g., pieces placed in two rows of different colours) from influencing the reception of the interface. Once entered the room, participants were informed that the session was going to be recorded and asked to sign a consent form, for which an ethical approval was previously requested and issued. We did not provide any explanation about the functionality and the objectives of the system. Participants were told that they could do whatever they wished and had no time limit.

At the end of each session, a researcher interviewed participants following a semistructured approach. Participants were free to talk about whatever they considered relevant to describe their experience. At the end of the interview,

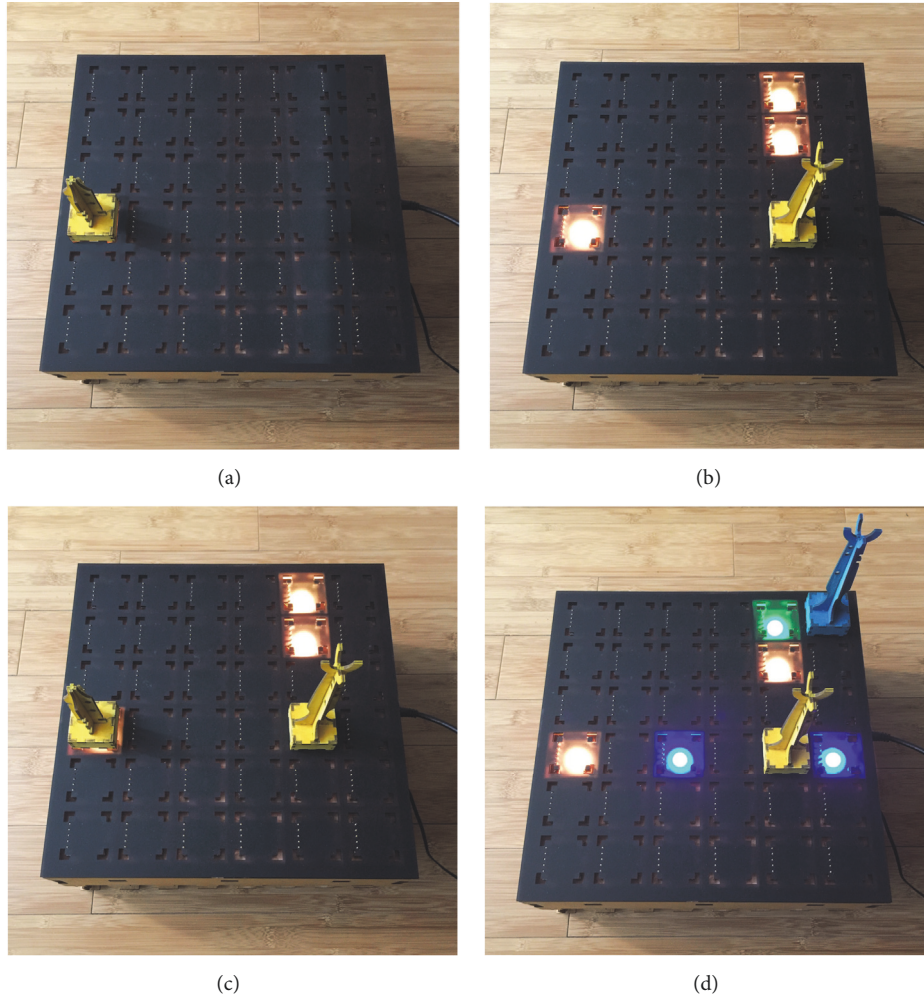


FIGURE 4: (a) Each cell is associated with a specific rhythm, which is played once a *pawn* is placed on top of it. (b) Once positioned on top of a cell, the *king* suggests favourable positions by illuminating up to three cells with its colour. The rhythm associated with these cells is similar to that where the *king* sits. (c) When a *pawn* is positioned on top of a cell with its same colour, its harmonic spectrum is enhanced. (d) When both *kings* are on the panel, their favourable positions might point to the same cell(s). In this case, a green colour lights up.

every participant was asked to describe the system they interacted with. Another source of data was provided by the videos; two researchers independently analysed the videos and thematically coded them. Coding was entirely data-driven and themes were derived in a number of ways, such as their prevalence in the data and also their importance. Double coding was conducted on around 30% of the videos, yielding interrater reliability of almost 90%.

A thematic analysis was performed on the data source composed of interview transcripts on the videos. A deductive approach was adopted: we had preexisting coding frames through whose lenses we aimed to read our research exploration [34]. The coding process identified the different behaviours exhibited by participants as well as the interpretations they attributed to the system. Then, a list of themes was identified by clustering codes. The following list indicates the identified themes and the associated codes.

- (i) *Visual exploration* {visual/light patterns; colour homogeneity; reaction to light; self-assignment of colours; presence of the pieces on top of the lights}
- (ii) *Sonic exploration* {timing in piece placement; awareness of sound source; awareness of drone sound; listening to sonic outcome; attempt to understand mapping}
- (iii) *Board exploration* {systematic exploration of each cell; geometric placement of pieces}
- (iv) *Pieces exploration* {moving pieces with no attention to sonic output; intentional movements of pieces; using the king; using the king only; piece rotation; awareness of colour and type of moved pieces}
- (v) *Interaction mode* {reaction to light; systematic exploration; movement combination; wait-and-see after a movement}

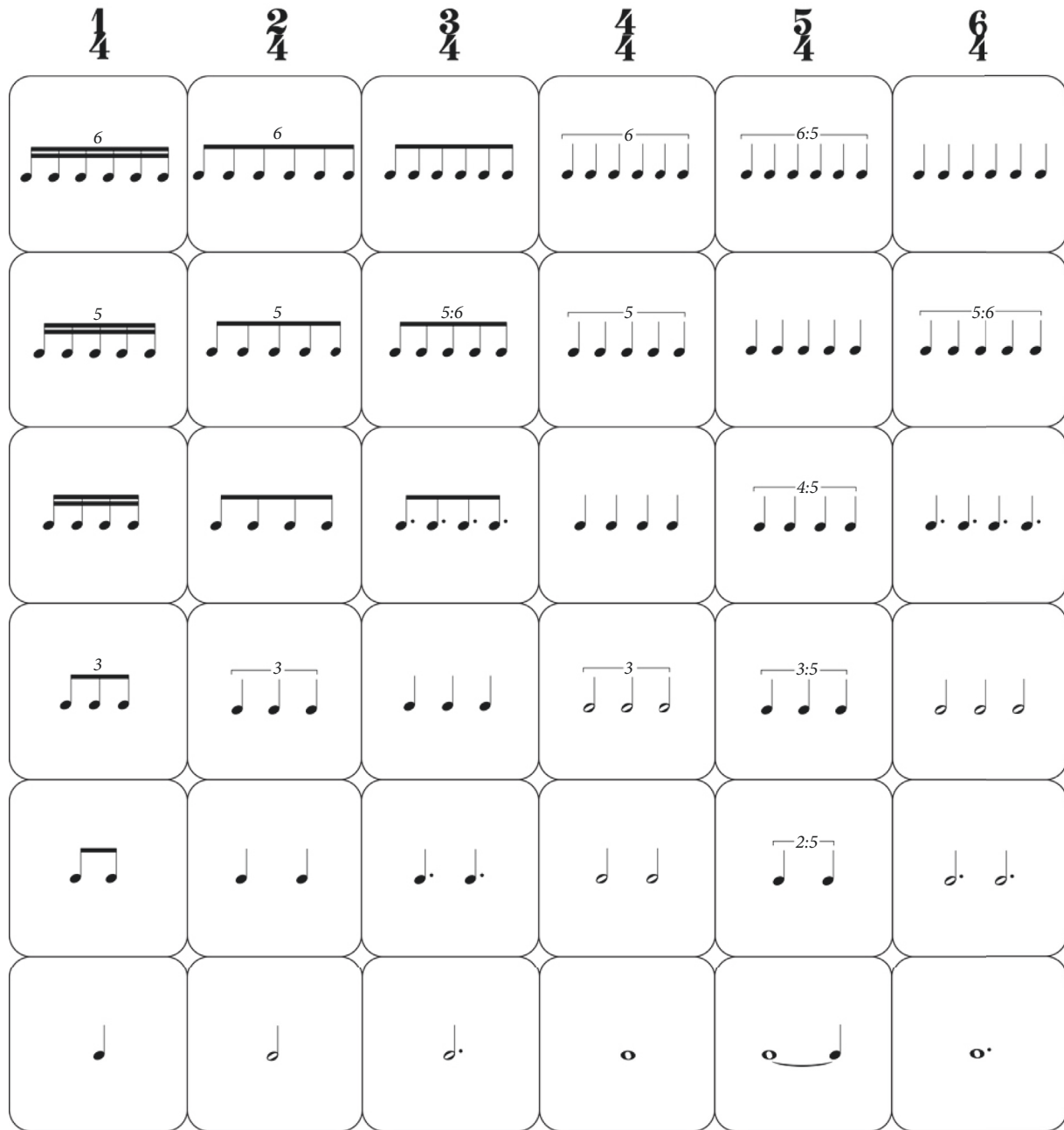


FIGURE 5: Table of possible rhythms on the matrix.

(vi) *Kind of engagement* {enjoyment; understanding mapping; epiphany/surprise; oral communication; involvement in music making; absorbed by the experience; appropriation of the system; strategic planning; flow of actions}

6. Results

Session lengths varied from 11.3 to 46.1 minutes (average 22.4, standard deviation 9.5). In general, we observed three different interpretations.

6.1. Exploration. Nearly all the participants received Beatfield as an exploratory game, but the objective of the exploration greatly varied. Our analysis shows that the *terminus of attention* was clustered into two almost orthogonal sets: on the one side, how the system functions; on the other side, what is the meaning of the system. Several participants spent most of the time trying to understand the functionality of the system, the possibilities they were offered, and the basic elements of the interaction. One group, composed of three art historians (all females, average age = 24), focused on understanding how the system worked, and how they could



FIGURE 6: The setup of Beatfield in the field study.

interact with it: “We spent most of the time trying to understand the underlying mechanism rather than purposely creating a melody or mixing different sounds”. Another participant (33, M, designer) reported: “I was trying to figure out what I could do with this ... thing”. Rather than focusing on the possibilities for interaction, some participants interpreted the system as sort of a puzzle game in which the objective was to understand the objective of the system. A board-game enthusiast (M, 33), for instance, said: “I tried to understand whether there was a predefined aim to this game. Because, apparently, it was a game. For instance, I tried nullifying sounds, or finding specific configuration for something special”. He also added: “I was trying to position the towers in a way that they could completely fill a specific area to see if something would happen”.

6.2. Immersion. In a few cases, players’ experiences resembled the psychological condition, called *flow* by Csikszentmihalyi, of being spontaneously absorbed in an activity [35]. In particular, an artist (F, 26), rather than trying a conscious interaction with the system, interpreted the system as a free exploration and she let herself be absorbed by it for more than 40 minutes. When prompted, she tried to explain her experience: “I was placing the pieces using my instinct. I believe you have to be free in this kind of experiences, you should not set yourself objectives”. She was not interested in understanding the values of the pieces: “I didn’t want to control the instrument too much. I always try to control everything, but this time I felt I was free, I let the music follow me. I realised that colours changed but I didn’t realise why it was that - I didn’t want

them to”. Rather, she connected Beatfield to her experience personal meanings and life events: “While playing I was having a sort of parallel existence. I was not here. All these cells, the slots for the towers. ...they were all connected to a picture I had in my mind. I lived through some moments of my life”.

6.3. Artistic Creation. Several participants interpreted Beatfield as a musical interface. Two professional musicians (F= 26, M=32), for instance, reported: “It is a controller. It reminded me of a loop station, and partially a sequencer. It is a way to create musical environments that can be more or less rhythmical. But with some randomness”. They reported that they had spent most of their session searching for polyrhythms, and for complex situations. However, although they spent more than 45 minutes interacting with the installation, they acknowledged that “it is too difficult to plan a precise musical interaction without knowing what’s behind the installation. Even if you had a musical objective in mind, reaching that configuration is virtually impossible. However, you can somehow create a musical landscape that moves”. In other cases, players focused their interaction on playing with visual elements, such as pieces’ placement and light combinations. As an example, two musicians, both expert chess players (24 and 26, M) spent a relevant amount of time trying to arrange the pieces into particular geometric shapes and to make the green lights appear: “When we had a lot of pieces on the surface we mostly played with the lights. And we also tried to play with geometry - perhaps a regular shape would create a particular sound”.

7. Findings

In this section we discuss the main findings of the work. First, we offer evidence that our proposed design model successfully elicited a variety of interpretations. Second, we reflect on how this model provides a new perspective for the design of interpretively flexible artefacts.

7.1. Fostering Multiple Interpretations. The field study demonstrated the capability of Beatfield to foster a range of possible interpretations and opportunities for interaction that have consequences on the very status of the system and on the experience of the player. Beatfield turned out to be a musical instrument, a board game, and an interactive artwork. Its perceived nature also changed throughout the sessions in several occasions. For instance, a passionate board-game player (21, F) initially fully dedicated her attention to controlling light activation, while she later tried to control the music towards the end of the session. The change in the interpretation of the system was mirrored by a change in the experience of the user, as explained by another participant (26, F, artist): “*At first, I was just placing towers on the board in complete freedom. I was not focusing on what I was doing. Then, I gradually listened to all the sounds and started to organise them*” to the point that she even found her favourite cell position: “*I really like the one that was playing there (position 6,6)*”.

The variety of interpretations disclosed different natures of the systems that were inherently embedded in it by the identities and values of the participants of the workshop, which strongly determined the meaning that users could make of Beatfield. The emergence of these different interpretations was favoured by the design activities, which gave voice to the different insights of the designers. These activities achieved to find a fine balance among the authors’ values, goals, insights, competences, and interests, which merged and grew together in a number of coauthored design scenarios that reflected the diversity of their proponents. The definition of these design scenarios was supported by the adoption of MINUET as a reference framework [14], whose design suggestions we operationalised. For instance, we proposed the idea of shuffling the order of the entities of the framework when conceptualising the new installation to allow different scenarios to emerge. This finding can appeal to readers interested in creating open-meaning artefacts and offers new knowledge on human interaction with multimodal devices.

It is worth mentioning that, even though Beatfield successfully elicited different interpretations, it also presented itself with a defined aesthetic feature as, especially from a musical perspective, the outcome is embedded in the artefact itself. We can therefore argue that Beatfield maintains some key elements of a *composed instrument*, given its intrinsic nature of being both a playable tool and a piece of music on its own [28]. However, Beatfield is not the product of a typical authored process but rather the product of a codesign process. This process also differs from that of the *cocreated composed tool*, proposed by Masu and Correia [29], as in that case the authorship over the final aesthetic musical results was negotiated between a composer/designer and a performer.

7.2. Coauthoring Interpretively Flexible Experiences. Following the conceptualisation of Sengers and colleagues [11], a system is considered *interpretively flexible* when its meaning or interpretation is coconstructed by the user and the designer. The model presented in this paper, which we schematically represent in Figure 7, departs from this view insofar as the coconstruction of meaning predates user interaction. The conditions for fostering multiple meanings are indeed set at design time by adopting design activities that embed the different values and ideas of the codesigners and that result in the emergence of different interpretations whose quality and number are as varied as the values and identities of the individuals that are involved. These design activities, which were inspired by the MINUET framework [14], allow the designer to set the general direction by offering initial suggestions (Table 1) as design material. Our investigation thus did not focus on evaluating the quality of the design suggestions against the experiences of the participants of our study. Rather, these suggestions constitute the very boundary of designer intervention.

8. Conclusion and Future Work

The expression “too many cooks in the kitchen” describes a situation in which an effort to achieve an outcome is made unproductive by too many individuals seeking to have input. In this paper, we advocate that this is not the case for multimodal interfaces that are designed to be interpretively open. We promoted the faculty of embedding in the design of a multimodal interface values and perspectives from different individuals to create heterogeneous interpretations. We proposed that this process benefits from being framed within a validated design framework and we tested this proposition with the case study of Beatfield, in which we intentionally avoided crafting the semiotics of the artefact to give rise to idiosyncratic interpretations of the system. Also, we expanded upon findings from related work to formulate design choices that could stimulate a variety of interpretations of an interactive system. Notably, this study did not precisely pinpoint the specific factors that contributed to such a diversity of experiences to emerge.

We also introduced the idea that the design of open-ended tools may benefit from nonobvious relations among the different elements of the artefact. In our case study, for example, different interpretations were elicited by having sound and visual independent from each other. The distinction among audio and visual elements can be considered *nonsyncretic*, which contrasts the syncretic approach, i.e., the amalgamation between audio and visuals in traditional media [37]. As opposed to syncretic approach to audiovisual, which advocate for the combination of audio and visuals to emphasise a concept, this approach purposely separates the two aspects to foster multiple interpretations. The results of the study confirmed that unclear relations between sound and visual indeed fostered different interpretations.

The findings of this paper open new challenges for multimedia devices. In particular, the conscious use of ambiguity in the design, relying on cocreation and adoption of nonobvious relations among different sensory modes, can

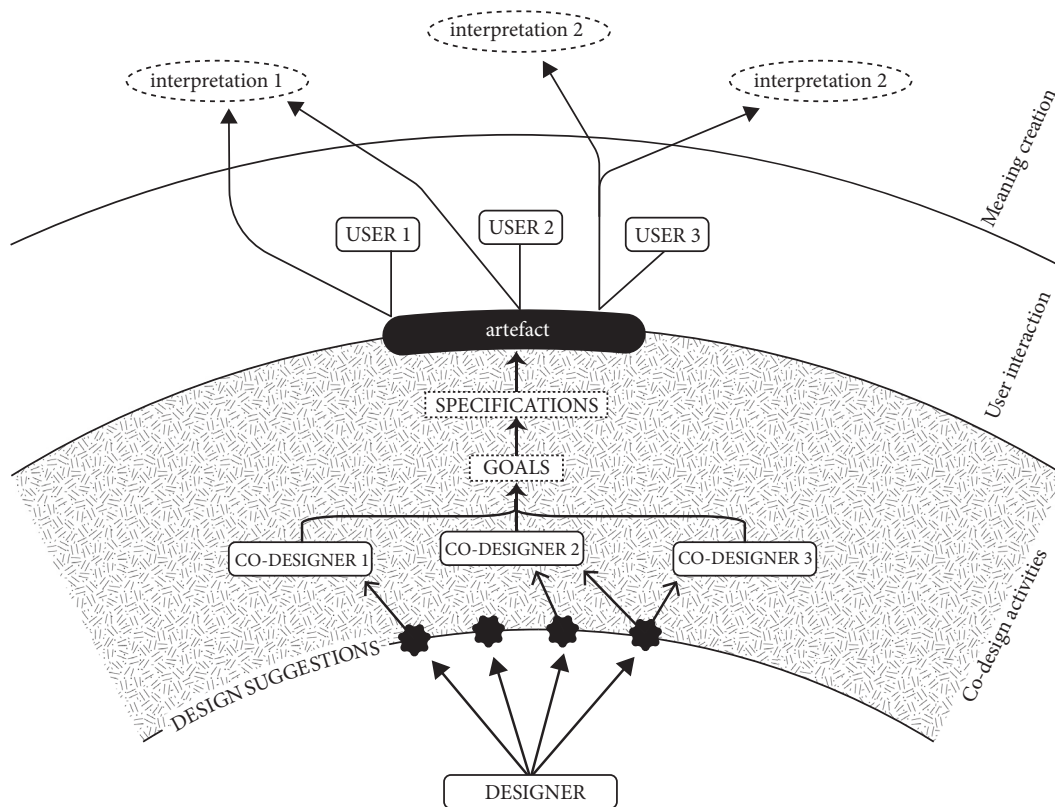


FIGURE 7: From the bottom-up: the designer initially outlines a series of *design suggestions* defining the constraints for user interaction. These suggestions are then used by codesigners during the codesign activities to frame the *goals* of the interaction and outline the *specifications* for the final artefact. At interaction stage, multiple *interpretations* of the artefact are stimulated.

help designers find solutions to make their artefact open to different interpretation, foster exploration, and support appropriation. Future studies will include a more copious number of participants to systematically test whether the emergence of specific interpretations and experiences is ascribable to the personal experience, interests, and background. Furthermore, future studies will explore differences between users experiencing the system individually or as a group and between participants tied by different forms of relationships. Finally, in this paper we focused on coauthorship at design time; future research will investigate issues of coauthorship in use when different people elaborate meaning by manipulating an ambiguous space together.

Data Availability

Interview transcripts and the videos of participants' sessions will be uploaded in an institutional website and freely accessible by everybody.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

We want to thank our colleagues that helped us with this project and in particular Zeno Menestrina, Andrea Conci,

and Adriano Siesser. We are also grateful to all the participants that volunteered to join our case study.

Supplementary Materials

A video collected during the field study is available as supplementary material. (*Supplementary Materials*)

References

- [1] V. Zappi and A. McPherson, "Dimensionality and appropriation in digital musical instrument design," *NIME*, pp. 455–460, 2014.
- [2] S. Reich, *Writings on music*, Oxford University Press, Oxford, UK, 2002.
- [3] F. Morreale and A. De Angeli, "Evaluating visitor experiences with interactive art," in *Proceedings of the 11th Biannual Conference on Italian SIGCHI Chapter*, pp. 50–57, ACM, 2015.
- [4] P. Dourish, "The appropriation of interactive technologies: some lessons from placeless documents," *Computer Supported Cooperative Work: CSCW: An International Journal*, vol. 12, no. 4, pp. 465–490, 2003.
- [5] A. Dix, "Designing for appropriation," in *Proceedings of the In Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI but not as we know it-Volume 2*, pp. 27–30, BCS Learning & Development Ltd., 2007.
- [6] S. Lindtner, S. Bardzell, and J. Bardzell, "Reconstituting the utopian vision of making: HCI after technosolutionism," in

- Proceedings of the 34th Annual Conference on Human Factors in Computing Systems, CHI 2016*, pp. 1390–1402, ACM, USA, May 2016.
- [7] A. L. Toombs, S. Bardzell, and J. Bardzell, “The proper care and feeding of hackerspaces: care ethics and cultures of making,” in *Proceedings of the 33rd Annual CHI Conference on Human Factors in Computing Systems, CHI 2015*, pp. 629–638, ACM, Republic of Korea, April 2015.
 - [8] F. Morreale, G. Moro, A. Chamberlain, S. Benford, and A. P. McPherson, “Building a maker community around an open hardware platform,” in *Proceedings of the 2017 ACM SIGCHI Conference on Human Factors in Computing Systems, CHI 2017*, pp. 6948–6959, ACM, USA, May 2017.
 - [9] S. Bødker, “Third-wave HCI, 10 years later - participation and sharing,” *Interactions*, vol. 22, no. 5, pp. 24–31, 2015.
 - [10] K. Kwastek, *Aesthetics of Interaction in Digital Art*, Mit Press, Cambridge, Massachusetts, USA, 2013.
 - [11] P. Sengere and B. Gaver, “Staying open to interpretation: engaging multiple meanings in design and evaluation,” in *Proceedings of the Conference on Designing Interactive Systems, DIS2006*, pp. 99–108, USA, June 2006.
 - [12] K. Höök, P. Sengers, and G. Andersson, “Sense and sensibility: evaluation and interactive art,” in *Proceedings of the The CHI 2003 New Horizons Conference Proceedings: Conference on Human Factors in Computing Systems*, pp. 241–248, USA, April 2003.
 - [13] W. W. Gaver, J. Bowers, A. Boucher et al., “The drift table: designing for ludic engagement,” in *CHI’04 Extended Abstracts on Human Factors in Computing Systems*, pp. 885–900, 2004.
 - [14] F. Morreale, A. De Angeli, and S. OModhrain, “Musical interface design: an experience-oriented framework,” in *Proc. NIME*, pp. 467–472, 2014.
 - [15] W. W. Gaver, J. Beaver, and S. Benford, “Ambiguity as a resource for design,” in *Proceedings of the The CHI 2003 New Horizons Conference Proceedings: Conference on Human Factors in Computing Systems*, pp. 233–240, USA, April 2003.
 - [16] G. Bell, M. Blythe, and P. Sengers, “Making by making strange: defamiliarization and the design of domestic technologies,” *ACM Transactions on Computer-Human Interactions (TOCHI)*, vol. 12, no. 2, pp. 149–173, 2005.
 - [17] C. Muth, V. M. Hesslinger, and C.-C. Carbon, “The appeal of challenge in the perception of art: how ambiguity, solvability of ambiguity, and the opportunity for insight affect appreciation,” *Psychology of Aesthetics, Creativity, and the Arts*, vol. 9, no. 206, 2015.
 - [18] T. W. Leong, “Designing for experiences: randomness as a resource,” in *Proceedings of the 6th conference on Designing Interactive systems*, pp. 346–347, University Park, PA, USA, June 2006.
 - [19] D. Birnbaum, R. Fiebrink, J. Malloch, and M. M. Wanderley, “Towards a dimension space for musical devices,” in *Proceedings of the In Proceedings of the 2005 conference on New interfaces for musical expression*, pp. 192–195, 2005.
 - [20] A. Johnston, L. Candy, and E. Edmonds, “Designing and evaluating virtual musical instruments: facilitating conversational user interaction,” *Design Studies*, vol. 29, no. 6, pp. 556–571, 2008.
 - [21] A. McPherson, “The magnetic resonator piano: electronic augmentation of an acoustic grand piano,” *Journal of New Music Research*, vol. 39, no. 3, pp. 189–202, 2010.
 - [22] J. Harrison, R. H. Jack, F. Morreale, and A. McPherson, “When is a guitar not a guitar? cultural form, input modality and expertise,” in *Proc. NIME*, 2018.
 - [23] J. Snyder, “Snyderphonics manta controller, a novel usb touch-controller,” in *Proc. NIME*, pp. 413–416, 2011.
 - [24] V. Zappi, A. Allen, and S. Fels, “Shader-based physical modelling for the design of massive digital musical instruments,” in *Proc. NIME*, pp. 145–150, 2017.
 - [25] F. Morreale, A. De Angeli, R. Masu, P. Rota, and N. Conci, “Collaborative creativity: the music room,” *Personal and Ubiquitous Computing*, vol. 18, no. 5, pp. 1187–1199, 2014.
 - [26] G. Lepri and A. McPherson, “Mirroring the past, from typewriting to interactive art: an approach to the re-design of a vintage technology,” in *Proceedings of the New Interfaces for Musical Expression*, NIME, Blacksburg, Virginia, USA, 2018.
 - [27] F. Morreale, A. McPherson, M. Wanderley et al., “Nime identity from the performer’s perspective,” in *Proc. NIME*, 2018.
 - [28] N. Schnell and M. Battier, “Introducing composed instruments, technical and musicological implications,” in *Proc. NIME*, pp. 1–5, National University of Singapore, 2002.
 - [29] R. Masu and N. N. Correia, “Penguin: design of a screen score interactive system,” in *Proceedings of the International Conference on Live InterFaces*, 2018.
 - [30] E. Aarseth, “I fought the law: transgressive play and the implied player,” in *From literature to cultural literacy*, pp. 180–188, Springer, 2014.
 - [31] B. Suits, *The Grasshopper: Games, Life and Utopia*, Broadview Press, 2014.
 - [32] B. De Koven, *The Well-Played Game: A Player’s Philosophy*, mit Press, 2013.
 - [33] Z. Menestrina, R. Masu, M. Bianchi, A. Conci, and A. Siesser, “OHR,” in *Proceedings of the 1st ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play, CHI PLAY 2014*, pp. 355–358, Canada, October 2014.
 - [34] V. Braun and V. Clarke, “Using thematic analysis in psychology,” *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, 2006.
 - [35] M. Csikszentmihalyi, “Toward a psychology of optimal experience,” in *Flow and the foundations of positive psychology*, pp. 209–226, Springer, 2014.
 - [36] J. Marshall, S. Benford, and T. Pridmore, “Deception and magic in collaborative interaction,” in *Proceedings of the 28th Annual CHI Conference on Human Factors in Computing Systems, CHI 2010*, pp. 567–576, USA, April 2010.
 - [37] M. Chion, C. Gorbman, W. Murch, and Audio-vision., *Audio-Vision*, 1994.

Research Article

Interaction Topologies in Mobile-Based Situated Networked Music Systems

Benjamin Matuszewski ^{1,2}, Norbert Schnell,³ and Frederic Bevilacqua²

¹CICM/musidance EAI572, Université Paris 8, Paris, France

²UMR STMS IRCAM-CNRS-Sorbonne Université, Paris, France

³Faculty of Digital Media, Furtwangen University, Germany

Correspondence should be addressed to Benjamin Matuszewski; benjamin.matuszewski@ircam.fr

Received 31 August 2018; Accepted 10 December 2018; Published 5 March 2019

Guest Editor: Stefania Serafin

Copyright © 2019 Benjamin Matuszewski et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we present a complete framework, both technical and conceptual, aimed at developing and analysing Networked Music Systems. After a short description of our technical framework called *soundworks*, a JavaScript library especially designed for collective music interaction using web browser of mobile phones, we introduce a new conceptual framework, we named *interaction topologies*, that aims at providing a generic tool for the description of interaction in such systems. Our proposition differs from the theoretical approaches generally proposed in the literature by decoupling the description of interaction topologies from the low level technical implementation of the network. We then report on a set of scenarios and prototypes, illustrating and assessing our framework, which were successfully deployed in several public installations and performances. We particularly show that our concept of interaction topologies succeeds at describing and analysing global aspects of interaction from multiple point of views (e.g., social, human-computer) by allowing for composing simple abstract figures. We finally introduce a discussion on agencies and perception of users engaged in such systems that could later complete our framework to conceive and analyse Networked Music Systems.

1. Introduction

The use of computer networks in music performances has a long history. The League of Automatic Composers in 1978 can be cited among the first ones, at least as one documented in the literature [1]. Since then, many tools and protocols necessary to transmit musical information have been developed to handle communication between devices, such as the ubiquitous MIDI [2] and OSC [3] protocols. In particular, the Internet infrastructure and its related protocols have enabled researchers and artists to create various *ad-hoc* music networks systems. More recently, the developments of web standards—with its numerous Application Programming Interfaces (APIs) [4] and its vast ecosystem of libraries—have significantly enhanced the rapid development of complex music network systems. As we will describe in this article, web technologies can be easily and favourably combined with mobile and miniature computer systems.

A number of authors have formalized musical computer networks [5–9] from a theoretical point of view and proposed various classifications. For example, Renwick in [9] has proposed a very broad definition of “Network music” as a “musical practice in which conceptual, technological, ideological, and/or philosophical concepts of the network are included in the design, composition, production, and/or performance process. The network may influence the work’s aesthetic, composition, production, or reception. The network may or may not be limited to electronic computerised networks”.

Several theoretical works on Network Music Systems have specifically focused on the case where users (performers, spectators) are located in different spaces. For example, Lazarro describes “*Network Musical Performance* (NMP) occur[ing] when a group of musicians, located at different physical locations, interact over a network to perform as they would if located in the same room” [10].

In this paper, we refer to the opposite cases where a group of people—small or large, expert or not—play, listen, and interact together in a shared *space and time*, systems that we qualify as *situated*. Therefore, we refer to our cases as Situated Networked Music Systems. Our approach is similar to Weinberg’s concept of *Interconnected Musical Networks* (IMN) [6, 11] that considers social interactions as key elements: “My definition for IMNs - *live performance systems that allow players to influence, share, and shape each other’s music in real-time* - suggests that the network should be interdependent and dynamic, and facilitate social interactions.” Furthermore, we propose to also consider any actor, human or not, performing or not, as a node in the network. Globally, according to the taxonomies proposed by Barbosa and Weinberg, every application we will describe here can be categorized as Local Musical Networks, Collective Creation Systems [5], or Real-Time Local Networks [6].

We implemented these concepts using mobile technologies (typically smartphone and tablet), relying on web standards—such as the relatively recent Web Audio API (<https://www.w3.org/TR/webaudio/> accessed 29 November 2018)—and Wi-Fi network capabilities.

Our main contributions are thus threefold. First, we will describe a conceptual and technical framework based on web technologies aimed at developing Situated Networked Music Systems. We will particularly introduce a conceptual framework dedicated to the description of such network systems, we call *interaction topologies*. This is in contrast to a technological point of view often reported in the literature that is solely based on low-level information network. Our approach allows for consolidating current theoretical frameworks by decoupling the topological description from its low-level technical implementation aspects. Second, we report on a set of Situated Networked Music Systems, implemented with our technical framework, that illustrate the use of the interaction topologies we propose. Finally, we introduce a discussion on perception and agencies that could offer a complementary perspective, centered on users, to the proposed conceptual framework.

2. Conceptual and Technical Framework

In this section, we describe a framework dedicated to Networked Music Systems. The framework is composed of two complementary components:

- (1) A technical part based on web standards—the *soundworks* framework—for rapid prototyping of collective and interactive scenarios, similar systems have been described in [12–14].
- (2) A theoretical part based on the concept of *interaction topologies*, aimed at describing and analysing such systems.

2.1. The Soundworks Framework. The scenarios of musical interaction explored in this research require that participants can spontaneously join an experience and interact within a distributed environment composed of numerous devices, such as smartphones. In order to enable short cycles in

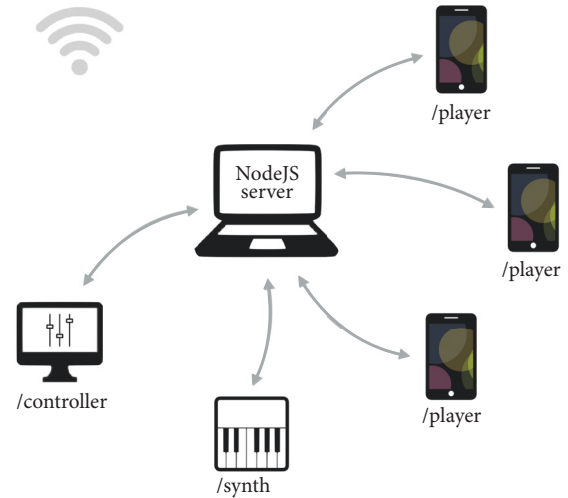


FIGURE 1: Network of interactive and synchronized audiovisual elements based on web technologies.

an iterative design process, the applications have to be rapidly prototyped. Moreover, they must be easily deployed to arbitrary audiences.

These constraints led us to create a prototyping environment based on web standards. Indeed, these technologies have the following qualities in our context:

- (i) Applications can be developed rapidly and immediately deployed on local or public networks.
- (ii) Participants can access applications with the web browser already installed on their smartphones connected through Wi-Fi or 3G.
- (iii) Web standards provide a number of APIs for interactive multimedia (e.g., audio synthesis, 2D and 3D rendering, motion sensors, geolocation), and real-time networking [4].

Furthermore, web technologies allow for easily integrating additional devices as clients of our system [15, 16], enabling for a wide range of interaction and audiovisual rendering possibilities. From this perspective, the scenarios we discuss here can be described as networks of interactive audiovisual elements that are dynamically constituted or completed by the mobile devices of the participating audience (cf. Figure 1).

To support experimentation of a wide range of different scenarios, we developed a JavaScript framework, *soundworks* (<https://github.com/collective-soundworks/soundworks> License BSD-3-Clause, accessed 29 November 2018), that provides a set of services and abstractions for the most common requirements and functionalities of such applications. The framework is entirely based on web standards on the client side and uses Node.js (<https://nodejs.org/en/> accessed 29 November 2018) on the server side. Since its very first version [17], the framework has been iteratively redesigned and became the basis of numerous applications.

A *soundworks* application typically consists of a set of synchronized web clients that connect to a server through a

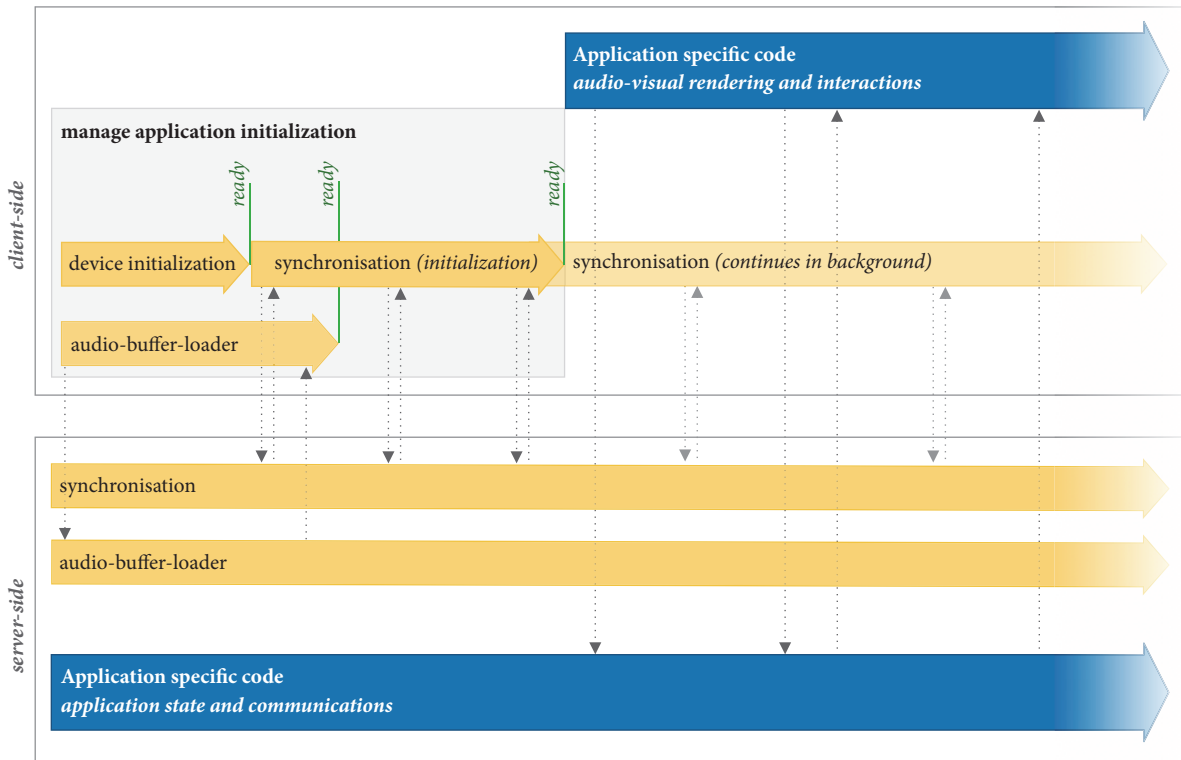


FIGURE 2: Initialization process of a typical *soundworks* application.

wireless network to exchange messages and data streams (see Figure 1). Depending on the context, an application may be deployed locally through a dedicated Wi-Fi network or over the Internet using existing Wi-Fi or 3G/4G infrastructures. The former allows for rapid iterations during development and test of an application, as well as a better control of bandwidth and latencies. However, the latter is more suitable for large scale events, especially outdoors.

The underlying philosophy of the framework is to provide a single place to write application specific code (i.e., the *Experience*), while being able to easily access predefined pieces of functionality (e.g., clock synchronization, preloading of sound files) by simply requiring a dedicated service. Among the numerous services and abstractions provided by the framework, the most important ones are:

- (i) clock synchronization between clients and server [18], similar to the Network Time Protocol (NTP) [19]
- (ii) real-time messaging and data streaming based on WebSockets
- (iii) shared parameters state between clients and server
- (iv) synchronized scheduling of (audio) events [20]
- (v) loading of sound files and related annotations
- (vi) simple abstractions for HTML views and 2D rendering.

Another important feature of the framework is also its ability to automatically manage the initialization of these interdependent processes—that may require communicating with the server to initialize (see Figure 2).

Figure 2 illustrates the initialization process of a typical application composed of a user-defined experience that uses three services dedicated to device initialization, synchronization (the *sync* service), and management of audio assets (the *audio-buffer-manager* service). The device initialization service that does not have any server-side counterpart is principally aimed at verifying that the client (e.g., smartphone's browser) supports all the APIs required by the application, and at resuming audio rendering when a user gesture (e.g., a *touch*) is captured. In parallel, the *audio-buffer-manager*—responsible for loading sound files and annotations from the server—starts to request sound files to the server. The *sync* service, on the contrary, relies on the audio clock to work, as such it must wait for the *ready* event of the *platform* service to start the synchronization process with the server. When all services have fired their *ready* event, the application specific code can start safely.

Alongside with the framework, an application template is also available (<https://github.com/collective-soundworks/soundworks-template> License BSD-3-Clause, accessed 29 November 2018). This template contains all the boilerplate code and generic configuration necessary to the framework. As such, it provides a simple and structured way of accessing

the APIs exposed by the framework and thus allows for starting the development of a new application in a few minutes.

2.2. Interaction Topologies. An important problem of existing approaches regarding analysis of topologies in Networked Music Systems comes with the idea that the “social organization of the network, an abstract, high-level notion, is addressed by designing and implementing the lower-level aspects of the network’s topology and architecture” [6]. Regarding our technical framework—which is entirely based on a centralized server—this statement would imply reciprocally the impossibility to design scenarios and interactions outside from a star topology (the *flower* topology in Weinberg’s terminology). To overcome this technological orientation concerning topologies, we introduce here the concept of *interaction topologies*. This approach aims at proposing a set of basis figures that can be used to describe several levels of interactions without focusing solely on technical aspects. As such, it proposes to describe networks of relations between entities (e.g., human, technical artifacts) without any *a priori* hierarchy on their agencies. Furthermore, the deliberate simplicity and genericity of the proposed graphs seek to promote their reuse for descriptions in multiple dimensions (e.g., time, space, and information flow) and, thus, emphasize the decoupling of the description of interactions from their underlying technical implementation. Indeed, while some of the abstractions provided by *soundworks* can support and ease the implementation of these figures in multiple ways, there is no one-to-one correspondence between the provided APIs and the figures presented here.

While numerous formal graphical notations dedicated at precisely modeling systems (or some of their components) have been proposed in the HCI and CS communities (e.g., petri nets, statecharts [21], or, more recently, the interface relational graph system [22]), our aim is to propose a complementary and high-level perspective that tries to stress the similarities between the described systems rather than their specificities.

Figure 3 shows the set of six figures—the *disconnected graph* (a), the *unidirectional circular graph* (b), the *bidirectional circular graph* (c), the *centrifugal star graph* (d), the *centripetal star graph* (e) and the *forest* (f)—that we propose. These graphs represent the actual possible interaction between each entity, human and technical artifacts. Importantly, they do not correspond to the representation of low-level information transmission through the network, as represented in Figure 1.

Our guess is that this minimal set could be sufficient for describing, analysing, and classifying Networked Music Systems from several perspectives. In the next section, we precisely describe a series of examples illustrating different interaction topologies.

3. Scenarios and Prototypes

In this section, we describe a set of experiments and scenarios that have been explored and refined during our research. The

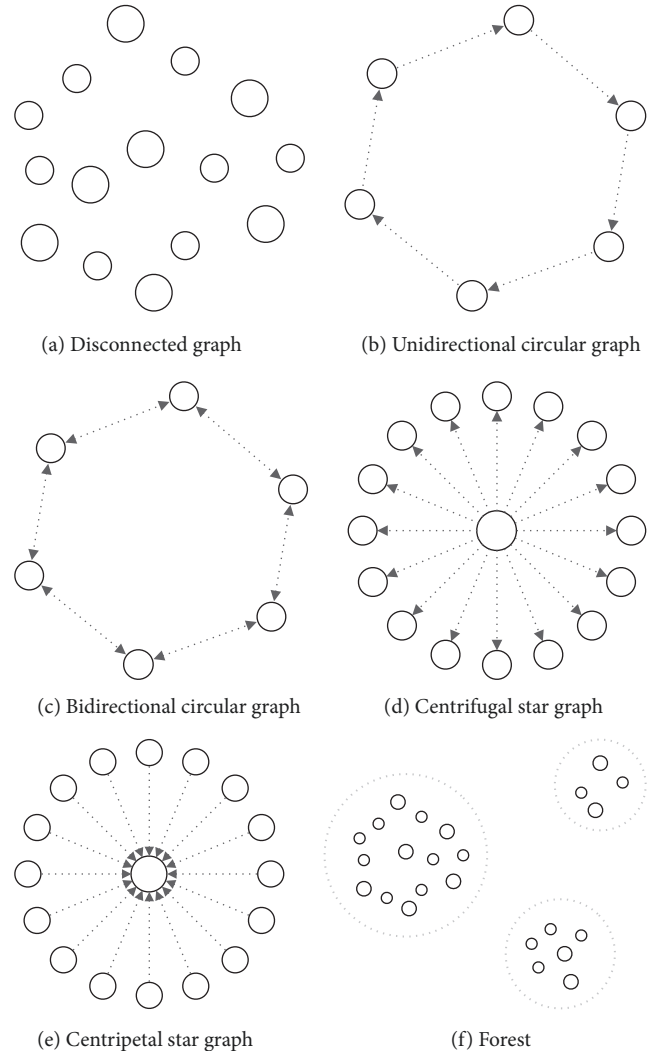


FIGURE 3: Abstract figures proposed a basis for the description of *interaction topologies*.

choice of the scenarios presented here also aims at illustrating each of the interaction topology figures presented in Section 1. As we will see, an interesting aspect concerning the proposed figures—and perhaps a byproduct of their simplicity—is the possibility to describe a single system from multiple points of views by combining several figures.

3.1. Birds, Rain Sticks, and Monks. This application proposes a set of simple gesture-controlled instruments. The main objective of the application is to propose to participants a didactical and ecological approach to create multisource sonic environments by giving them access to simple instruments created around obvious metaphors (e.g., rain stick) and gesture interactions based on motion sensors (e.g., shake, orientation). Once the application is loaded on the web browser of each mobile, every participant can entirely act independently, corresponding thus to the *disconnected graph* topology (see Figure 4).



FIGURE 4: *Birds, Rain Sticks, and Monks* at *Paris Face Cachée*, Ircam, Paris, 2015. The application has been first tested in a series of workshops conducted with Studio 13/16 at the Centre Georges Pompidou (Paris, France) in Spring 2014 and Fall/Winter 2014.

3.2. *Drops*. The *Drops* experience has been strongly inspired by the iOS application *Bloom* developed by Brian Eno and Peter Chilvers (<http://www.generativemusic.com/bloom.html> accessed 29 November 2018). Similarly to the *Bloom* application, the *Drops* application allows players to touch the screen of their mobile device to generate *drops*. Each generated *drop* is characterized by two complementary aspects of rendering: the trigger of a percussive and resonant sound through the device loudspeakers and a colored circle that grows at the touch position and fades away. According to the touch position, users can control simultaneously the pitch and the duration of the sound.

Unlike *Bloom*, *Drops* has been designed for an unlimited number of colocated participants playing together. The participants' mobile devices are synchronized and each drop played by a participant is echoed on the device of two other participants' devices before coming back to the original device. The delay and attenuation introduced in each echo produce an ever evolving and vanishing distributed texture among the participants. Additionally, each participant is

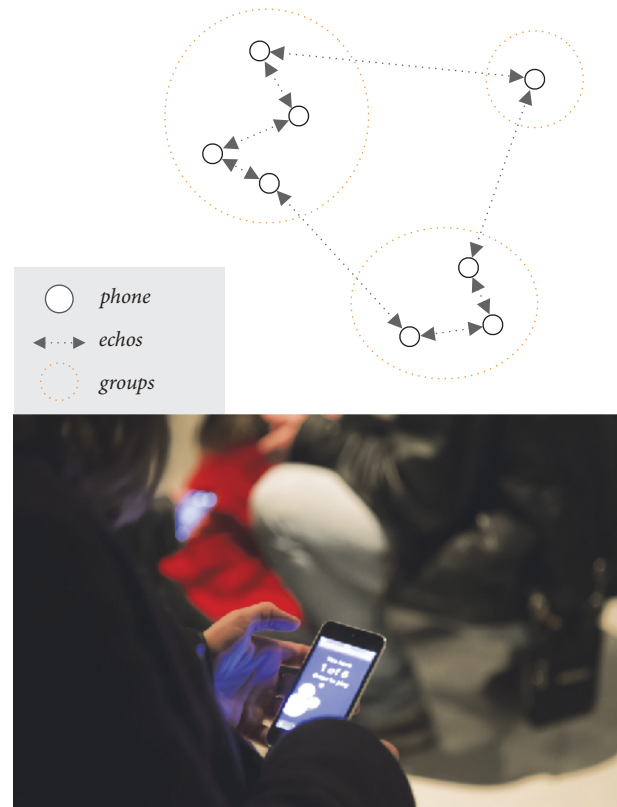


FIGURE 5: *Drops* at *Paris Face Cachée*, Ircam, Paris, 2015. The first performances of *Drops* occurred during a series of workshops conducted at Ircam (Paris, France) on February 7th 2015, in the context of the 4th edition of the *Paris Face Cachée* event.

associated with a specific color that allows for identifying his contributions as well as other participants' contributions on his own screen.

In its situated version, where all the people are in the same location, *Drops* can be first described as a *bidirectional circular graph* topology.

We also created an online version of the application, where each participant is geolocalized and relationships between participants are created by minimizing the distance between each of them (i.e., an application of the salesman problem). For example, participants who are close are grouped and can play together. Persons of this subgroup will still remain connected to people in other groups located elsewhere. As shown in Figure 5, this can be described by superimposing the *forest* figure to the *bidirectional circular graph*.

3.3. *Collective Loops*. *Collective Loops* is an installation that allows up to eight users to collaboratively interact within a shared audio-visual environment using handheld devices [15]. Conceptually, the whole installation can be seen as an 8-step loop sequencer in which each participant embodies a single step of the sequence.

The installation is composed of two different interleaved and synchronized layers. At the local level, a participant's

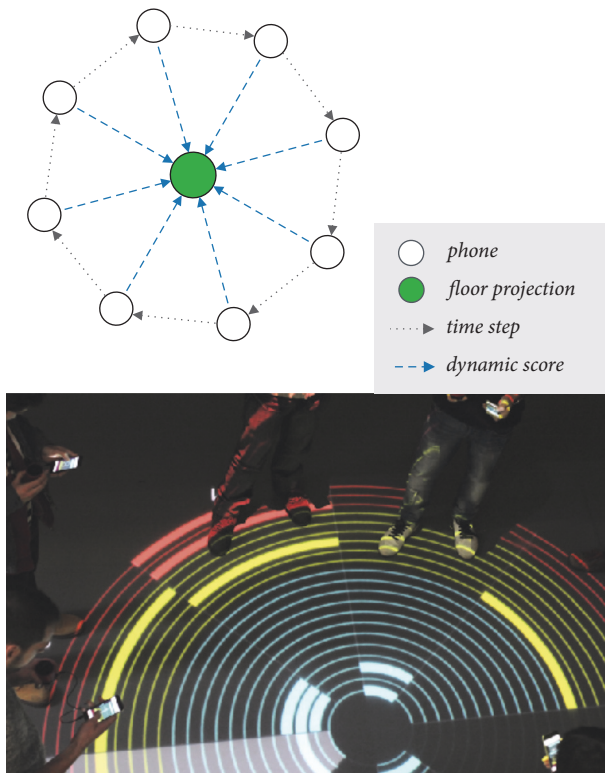


FIGURE 6: *Collective Loops* at Ircam Forum Workshop, Paris, 2015. The second version of *Collective Loops* described here has been first showcased in the context of the *Ircam Forum Workshops* that took place in November 25th and 26th, 2015, at Ircam (Paris, France). <https://vimeo.com/149649477> accessed 29 November 2018.

mobile device acts both as a controller and as an audio source. As a controller, the device exposes two different modalities of interactions: the user can enable and disable particular notes of its synthesizer by touching the screen but also control the cutoff frequency of a lowpass filter by modifying the position of the device around its pitch axis.

Additional to the participants' devices, the installation features a shared visual projection on the floor that reproduces the current state of the control of all devices. In addition to provide a way to place the participants in space, this shared representation also allows for enhancing collaborative aspects by giving participants information on one another's actions. Finally, the projection offers a simple way to follow the advance of the step in the sequence.

As a form of step sequencer and from the point of view of the control and audio rendering, *Collective Loops* can be described as a *unidirectional circular graph* where a token advances according to a predefined time step. However, by considering the system from the point of view of the shared visual rendering projected on the floor—on which every participant contributes equally—we can as well describe the system as a *centripetal star* (see Figure 6). Furthermore, this shared rendering also creates a new way for participants to interact with one another (e.g., by creating visual figures such as circles or stairs). From this point of view, the system could also be seen as a *centrifugal star* topology. Hence, several

layers of intertwined audio, visual, and social interactions could be described by the simple combination of three basis figures.

3.4. GrainField. *GrainField* requires the presence of an improvising performer (instrumentalist or singer) placed in the center of the audience seated around her/him.

In this experience, the performer is continually recorded by the system which creates every second an audio file of the two previous seconds of recording. This process occurs continuously from the beginning to the end of the improvisation. Each time a new audio file is created, it is sent to a random selection of the participants' mobile (representing typically 10% of the audience). On the participants' mobiles side, the received sound files are replayed in a granular synthesizer. Participants can scrub into the samples by waving their device to control the playback position. The screen of the device is only used to give additional feedback to users in two different ways: by displaying the current playback position of the synthesizer and by changing the background color each time a new sample is received.

Another client of the system—that is not seen by participants—allows for globally controlling synthesis parameters (e.g., grain duration, resampling) on every participant's devices, in order to adapt to and/or reinforce some characteristics of the performance. The resulting global audio rendering can be described as a distributed granular echo of the sound material proposed by the performer, creating an ever evolving texture.

In term of topology, as the musical material created by the performer is distributed over the participants smartphones, the system can be described as a *centrifugal star* (see Figure 7). However, the performer is also influenced by the feedback received from this delayed and granularized material; therefore, from this other point of view, the topological description of the system could also be complemented by a *centripetal star*.

3.5. 88 Fingers. *88 Fingers* is a collaborative performance in which up to 88 participants perform on an automatized piano (i.e., a YAMAHA Disklavier) using their mobile devices. The performance plays with codes of the classical concert by keeping the scenography of a piano recital: the piano on stage while participants sit in the room.

At the beginning of the performance, each participant can choose one single key of the piano among the remaining ones (once a key has been chosen by a participant, it is no longer available for others). When the performance starts, participants can play their key for the duration of the performance by simply touching the screen of their mobile phone. The graphical interface allows for controlling only two parameters: pressing the key of the piano by touching the screen with a velocity that corresponds to the vertical position of the touch.

The experience is built around ideas of “freedom and responsibility,” by not adding any additional rules to the system (computational or verbal). From an interaction point of view, it corresponds to the *centripetal star graph*, where



FIGURE 7: *GrainField* at the *AudioMostly* conference, London, 2017. Performance by Peyman Heydarian, *Santur*. The first performance of *Grainfield* took place on May 26th 2016 in Berlin (Germany) in the context of the *Hack the Audience* workshop organised during the *Music Tech Fest* event. This first version has been designed together with and performed by Takumi Motokawa and Karl Pannek.

each participant acts towards a single element, the piano (see Figure 8). Hence, it can be seen as the reverse of *GrainField* in terms of interaction topology.

3.6. ProXoMix. *ProXoMix* is an installation where participants, equipped with mobile devices connected to earphones, interactively remix a piece composed of complementary loops by moving physically in the space. In this installation, each participant embodies a predefined track that can be chosen through a dedicated interface. Once inside a track, the participant can modulate its content with two complementary modalities: samples composing the track can be activated and deactivated by touching the screen and the cutoff frequency of a lowpass filter can be changed by tilting the device.

The principal interaction, however, consists in moving in the space to get closer to other participants. Indeed, when two or more participants get close enough from one another, they start to hear the track of their peers with their earphones. The installation engages participants to collaboratively mix the proposed tracks, creating thus social-musical assemblies and spontaneous choreographies.

In terms of topologies, the formation of small groups can be first seen as a *forest* topology. Nevertheless, the evolving forest that describes *ProXoMix* at the highest level can also

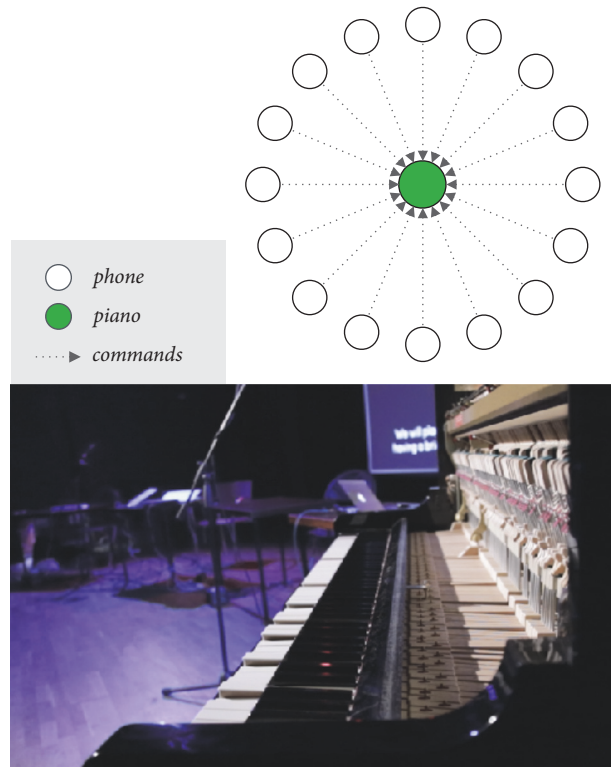


FIGURE 8: *88 Fingers* at the *AudioMostly* conference, London, 2017. The first performance of *88 Fingers* has been held in the context of a cultural event organised by the *JEP-TALN-RECITAL Conference* on July 7th 2016 at Centre Georges Pompidou (Paris, France).

be refined further. Indeed, each subgroup of the forest is characterized by the fact that all participants share the same audio rendering, creating thus the forest of centripetal stars illustrated in Figure 9.

4. Discussion and Conclusion

First, the examples we described were all shown several times, in different settings from public installations to concerts and performances. These public events demonstrated that each described system worked as planned from a technical point of view, and that the setup could directly scale for performances with up to 150 participants.

The interaction topologies we proposed offer one point of view we found useful for describing global view of interaction between the different elements of the systems, both human and technical. Such an approach is complementary compared to other approaches proposed in the literature [6].

Nevertheless, it is also interesting to assess a point of view based on the user experience. For this, we propose to take into account several properties, considering user degrees of freedom for action with the device and the musical constraints on the user actions, as well as the user perceived interaction and agencies. Table 1 shows a possible analysis for each of the proposed scenarios. The rating proposed here are based on our own experience of the systems as designers

TABLE 1: Degree of freedom, constraints, perception and agencies from a user point of view. These ratings are proposed by the authors as starting point for discussion.

<i>agencies</i>	<i>Birds</i>	<i>Drops</i>	<i>Collective Loops</i>	<i>GrainField</i>	<i>88 Fingers</i>	<i>ProXoMix</i>
degree of freedom of the interface	<i>low</i>	<i>med</i>	<i>med</i>	<i>low</i>	<i>low</i>	<i>high</i>
constraints on user actions	<i>low</i>	<i>med</i>	<i>high</i>	<i>low</i>	<i>low</i>	<i>low / med</i>
perceived personal agency	<i>high</i>	<i>high</i>	<i>med</i>	<i>low</i>	<i>low / med</i>	<i>med / high</i>
perceived contribution to shared rendering	<i>high / med</i>	<i>med / low</i>	<i>high</i>	<i>high</i>	<i>low / med</i>	<i>med / high</i>
perceived interaction with others	<i>med / low</i>	<i>med / low</i>	<i>high</i>	<i>med</i>	<i>low / med</i>	<i>high</i>

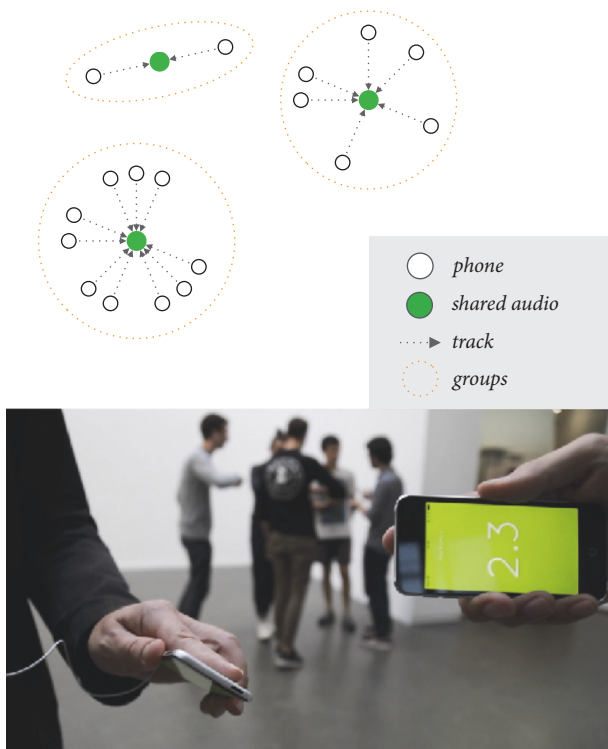


FIGURE 9: *ProXoMix* at the ESBA, Le Mans, 2016. The first version of *ProXoMix* has been showcased on September 22th 2016 in the context of a *CoSiMa* workshop organised by the ESBA Le Mans (France). <https://www.youtube.com/watch?v=a7CLfatCUNY> accessed 29 November 2018.

and discussions with participants. As such, our point here is not to propose a formal user evaluation for each, which would be out of the scope of this paper (interested readers can refer to [23] for such work concerning *Collective Loops*), but rather to propose a series of starting points for discussion and analysis.

These criteria provide complementary properties to the ones exhibited by the *interaction topologies*. The table illustrates that these criteria can also be used to distinguish between the different applications.

For example, in *Birds* and *Drops*, the systems have been designed to scale from small to large participating audience without technical modifications. We observed a shift in the way participants engage into the experience. Indeed, the perceived contributions to shared rendering and perceived interaction with other participants decrease as the number of participants engaged in the experience increases.

In *88 Fingers* and *ProXoMix*, the perceived agencies might vary depending on musical materials. For example, very low or very high pitches are much easier to perceive in the collective improvisation in *88 Fingers*. Similarly, some tracks in *ProXoMix*, such as drums or melodic tracks, are easier to perceive and have more musical impact compared to more discreet sound elements.

Interestingly, *GrainField* offers an example where the perceived contributions to the shared rendering remain clear while the possibilities offered by the system are really limited.

In summary, we have proposed a complete framework, both technical (open-source *soundworks* library) and conceptual (*interaction topologies*), aimed at developing and analysing Situated Networked Music Systems. We then presented a set of scenarios and prototypes developed using *soundworks* that illustrated each of the proposed topologies. Furthermore, we showed that the framework dedicated to *interaction topologies* can be used to describe our applications from multiple perspectives, confirming its qualities compared to approaches centered on technical aspects of the network topologies. We believe that the simplicity of the proposed approach and figures, which allows for combining several figures to describe a single application, could provide a powerful tool to describe and analyse a wider range of Networked Music Systems.

Still, a number of aspects of the framework can still be improved. On the technical side, while our platform has proved to be efficient for prototyping a wide range of application, an important work is currently performed to improve its accessibility to nonexpert programmers such as researchers and artists. On the theoretical side, and particularly concerning *interaction topologies*, these concepts should be assessed on a wider range of scenarios and applications. Also, a complementary work could be pursued to combine

our concept of interaction topologies with a user perspective taking into account interaction perception and agencies.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work described in this article has been conducted in the context of the CoSiMa research project funded by the French National Research Agency (ANR) [ANR-13-CORD-0010]. We would like to thank our colleagues Jean-Philippe Lambert, Sébastien Robaszkiewicz, David Poirier-Quinot, and Victor Audouze, as well as our partners Orbe, EnsadLab, ESBA TALM–Le Mans, ID Scènes, and NoDesign, for their valuable contributions to this work.

References

- [1] J. Bischoff, R. Gold, and J. Horton, "Music for an Interactive Network of Microcomputers," *Computer Music Journal*, vol. 2, no. 3, p. 24, 1978.
- [2] MIDI Manufacturers Association, *The complete MIDI 1.0 detailed specification*, The MIDI Manufacturers Association, Los Angeles, CA, USA, 1996.
- [3] M. Wright, *Open sound control 1.0 specification*, Center For New Music and Audio Technology (CNMAT), UC Berkeley, 2002.
- [4] L. Wyse and S. Subramanian, "The Viability of the Web Browser as a Computer Music Platform," *Computer Music Journal*, vol. 37, no. 4, pp. 10–23, 2013.
- [5] Á. Barbosa, "Displaced Soundscapes: A Survey of Network Systems for Music and Sonic Art Creation," *Leonardo Music Journal*, vol. 13, pp. 53–59, 2003.
- [6] G. Weinberg, "Interconnected Musical Networks: Toward a Theoretical Framework," *Computer Music Journal*, vol. 29, no. 2, pp. 23–39, 2005.
- [7] L. Gabrielli and S. Squartini, "Networked Music Performance," in *Wireless Networked Music Performance*, SpringerBriefs in Electrical and Computer Engineering, pp. 3–19, Springer Singapore, Singapore, 2016.
- [8] I. Hattwick and M. M. Wanderley, "A Dimension Space for Evaluating Collaborative Musical Performance Systems," in *Proceedings of the 2012 International Conference on New Interfaces for Musical Expression*, vol. 4, University of Michigan, 2012.
- [9] R. Renwick, *Topologies for Network Music*, [Doctoral thesis], Queens University, 2017.
- [10] J. Lazzaro and J. Wawrzyniek, "A Case for Network Musical Performance," in *Proceedings of the 2001 Conference on Network and Operating System Support for Digital Audio and Video*, pp. 157–166, ACM Press, 2001.
- [11] G. Weinberg, "The Aesthetics, History, and Future Challenges of Interconnected Music Networks," in *Proceedings of the 2002 International Computer Music Conference*, 2002.
- [12] L. Wyse and N. Mitani, "Bridges for Networked Musical Ensembles," in *Proceedings of the 2019 International Computer Music Conference*, 2009.
- [13] N. Weitzner, J. Freeman, Y. Chen, and S. Garrett, "massMobile: towards a flexible framework for large-scale participatory collaborations in live performances," *Organised Sound*, vol. 18, no. 01, pp. 30–42, 2013.
- [14] J. T. Allison, Y. Oh, and B. Taylor, "NEXUS: Collaborative Performance for the Masses, Handling Instrument Interface Distribution through the Web," in *Proceedings of the 2013 International Conference on New Interfaces for Musical Expression*, 2013.
- [15] N. Schnell, B. Matuszewski, J.-P. Lambert et al., "Collective Loops — Multimodal Interactions Through Co-Located Mobile Devices and Synchronized Audiovisual Rendering Based on Web Standards," in *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*, ACM, Yokohama, Japan, 2017.
- [16] B. Matuszewski and F. Bevilacqua, "Toward a Web of Audio Things," in *Proceedings of the 2018 Sound and Music Computing Conference*, Cyprus, 2018.
- [17] S. Robaszkiewicz and N. Schnell, "Soundworks—a Playground for Artists and Developers to Create Collaborative Mobile Web Performances," in *Proceedings of the 1st Web Audio Conference*, Paris, France, 2015.
- [18] J. P. Lambert, S. Robaszkiewicz, and N. Schnell, "Synchronisation for Distributed Audio Rendering over Heterogeneous Devices, in HTML5," in *Proceedings of the 2nd Web Audio Conference*, Atlanta, US, 2016.
- [19] D. L. Mills, "Internet time synchronization: the network time protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.
- [20] N. Schnell, V. Saiz, K. Barkati, and S. Goldszmidt, "Of Time Engines and Masters – An API for Scheduling and Synchronizing the Generation and Playback of Event Sequences and Media Streams for the Web Audio API," in *Proceedings of the 1st Web Audio Conference*, Paris, France, 2015.
- [21] D. Harel, "Statecharts: a visual formalism for complex systems," *Science of Computer Programming*, vol. 8, no. 3, pp. 231–274, 1987.
- [22] O. Mubarak, D. Bihanic, P. Cubaud, and S. Bianchini, "Art Installations: A Study of the Topology of Collective Co-located Interactions," in *Proceedings of the 8th International Conference on Digital Arts, ARTECH 2017*, pp. 23–30, ACM Press, New York, NY, USA, September 2017.
- [23] O. Mubarak, P. Cubaud, D. Bihanic, and S. Bianchini, "Designing Collaborative Co-Located Interaction for an Artistic Installation," in *Proceedings of the 2017 INTERACT International Conference*, pp. 223–231, Mumbai, India, 2017.

Research Article

Virtual Net: A Decentralized Architecture for Interaction in Mobile Virtual Worlds

Bingqing Shen  and **Jingzhi Guo** 

Faculty of Science and Technology, University of Macau, Taipa, Macau

Correspondence should be addressed to Jingzhi Guo; jzguo@umac.mo

Received 23 July 2018; Revised 21 October 2018; Accepted 30 October 2018; Published 8 November 2018

Guest Editor: Federico Fontana

Copyright © 2018 Bingqing Shen and Jingzhi Guo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of mobile technology, mobile virtual worlds have attracted massive users. To improve scalability, a peer-to-peer virtual world provides the solution to accommodate more users without increasing hardware investment. In mobile settings, however, existing P2P solutions are not applicable due to the unreliability of mobile devices and the instability of mobile networks. To address the issue, a novel infrastructure model, called Virtual Net, is proposed to provide fault-tolerance in managing user content and object state. In this paper, the key problem, namely, object state update, is resolved to maintain state consistency and high interaction responsiveness. This work is important in implementing a scalable mobile virtual world.

1. Introduction

Virtual worlds, including multiplayer online games and virtual social worlds, allow users to inhabit in virtual environments, create their own content, and interact with each other. Mobile virtual worlds allow users to access the simulated environments through mobile devices, achieving the possibility to play anywhere. Mobile virtual worlds have gained large attraction from the development of mobile devices. They have become an important market and revenue source for the game industry and attracted a large number of users. For example, Fortnite has earned \$1,996,917 gross daily revenue [1] and reported 3.4 million concurrent players [2] in 2018. The success and expansion of mobile virtual worlds raise new challenges in infrastructure development; one of them is the scalability problem. In virtual worlds, interaction is implemented by sending events to servers for processing and receiving updates from the servers for rendering and state synchronization. With the increase of concurrent online users, more computing load is imposed on game infrastructures. Servers have to process and respond to more client requests within a short period for high responsiveness. Also, network bandwidth consumption is increased to pack multiple game states in an update. For scaling, more

computing resources have to be invested. Otherwise, user experience will be affected.

Peer-to-peer (P2P) virtual worlds, firstly introduced in [3], explore the possibility of running a virtual world without a central server. In P2P virtual worlds, user devices run both the client program and server program for event handling and state update. Thus, computing resources naturally scale along with the change of user population. Mobile applications, however, have different characteristics with respect to their desktop counterparts. One outstanding issue is client failure. Compared to desktop PCs, mobile devices are more prone to failure, due to, for example, battery depletion or application conflict. Moreover, the access to mobile networks, such as MANETs and VANETs, is also unstable. Client unreliability may cause content loss or state inconsistency, if user content and object state are not properly saved or backed up before failure. Yet, existing P2P virtual worlds do not concern the peer device unreliability problem [4]. Thus, they cannot be directly applied in mobile settings.

In this paper, a Virtual Net model is proposed to address the client unreliability problem for mobile P2P virtual worlds. The model utilizes the cloud-fog structure, but totally decentralized. To avoid content loss, the cloud layer stores user contents for content persistency. The fog layer caches object

states for client recovery and maintains state consistency. The separation of content storage and state caching can improve responsiveness, since operations direct on P2P storage have more communication overhead [5]. Based on the P2P content storage, a content addressing scheme is devised, which can facilitate content integrity check.

To avoid reinventing the wheel, this paper mainly focuses on the state update problem to maintain object state consistency. At the fog layer, object states are replicated on several nodes for fault-tolerance. Thus, all replicas must maintain the same state in event handling so that interaction can be performed within a consistent shared environment. Yet, the requirement of high responsiveness in virtual world interaction makes the problem difficult. To attack the difficulty, an opportunistic approach, called fast event delivery, is proposed. Based on the approach, a virtual world interaction model is then designed. In short, the main contributions of the paper are listed as follows.

- (1) A new P2P cloud-fog structure, called Virtual Net model, is proposed to resolve the client unreliability problem, which can provide fault-tolerance in playing a mobile virtual world.
- (2) A fast event delivery approach is proposed to maintain both replica state consistency and high responsiveness in the process of handling user events.
- (3) A new virtual world interaction model is designed to achieve game state consistency and high responsiveness when interacting with different neighbors.

The remainder of the paper is organized as follows. The related works are introduced in Section 2. The overall Virtual Net model is described in Section 3. Section 4 studies the state update problem in detail. Based on the solution of the problem, the virtual world interaction model is provided in Section 5 with neighbor change management. The correctness of the solution is proved in Section 6. Sections 7 and 8 evaluate the performance through theoretical analysis and experiments. Section 9 concludes the paper.

2. Related Work

Mobile P2P virtual worlds combine the characteristics of mobile virtual world and P2P virtual world problems. Due to the lack of study in this field, the related work in P2P virtual worlds and cloud-fog mobile applications is surveyed to shape the distinct characteristics of the combined problem.

2.1. P2P Virtual Worlds. P2P MMORPGs and P2P virtual environments have been amply surveyed in [4, 6]. Previous works mainly focus on inter-player consistency management, including peer connectivity, interest management, event dissemination, and cheat prevention. Peer connectivity [7] studies the connection of all user devices within an overlay network such that any peer can be reached from another peer. Interest management [8] restricts the range of message receipt to reduce communication overhead in state update. Event dissemination [9] reduces the number of communication channels on event senders to avoid overwhelming them in

hotspot areas. Cheat prevention [10] is needed to achieve fairness without the arbitration from a central server. In these works, a desktop environment is assumed such that a client is always reliable in storage and connection. In contrast, the Virtual Net targets the mobile environments in which both devices and connections are unreliable, which is the new problem and orthogonal to the above studies. Thus, a complete implementation of Virtual Net can employ existing P2P solutions in inter-player consistency management, such as peer connectivity and interest management, to avoid reinventing the wheel.

Early work on P2P state persistency is related to the content storage in this work. State persistency studies the reliable storage and efficient retrieval of user state [11]. Each time a state is updated, it has to be persisted in the overlay network, and the state has to be queried from the overlay network when the client is recovered from a failure. Same as the above argument, the work in [11] only assumes a reliable client, which is not applicable in a mobile setting. The Virtual Net model not only solves the unreliable client problem, but also reduces storage and retrieval overhead through content caching. Moreover, content integrity check is included in Virtual Net, which is not mentioned in previous works.

2.2. Fog Computing. Firstly introduced in [12], cloud gaming moves the game engine functions to the cloud to simplify development, distribution, access, and update [13]. However, the measurement study [14] shows that the current cloud gaming infrastructure is unable to meet the latency requirement for end-users distant from data centers. To improve latency, fog computing [15] has been introduced to move the time-critical functions to the locations near clients. Fog computing has been widely discussed in both Internet of Things (IoT) [16] and mobile computing [17] to offload server burden [18], enable location awareness, and provide real-time interaction. Among its many applications, mobile gaming [19] and mobile reality [20] are two important examples. Similar to the cloud-fog structure, the Virtual Net solution also employs the cloud layer for content storage and the fog layer for latency improvement. But differently, Virtual Net explores a totally decentralized solution, with no central control at the cloud layer.

3. Virtual Net Model

The proposed Virtual Net structure is based on the commonly used three-layer structure shown in Figure 1. Similar to some existing cloud-fog structures, it is divided into three layers: the cloud layer (L1), the fog layer (L2), and the client layer (L3). The cloud layer provides persistency service, which stores the files of user content and the state of virtual objects (avatars, accessories, achievements, etc.). The fog layer caches object states in play and provides state recovery for clients in case of short-term failure. It also periodically checks object states and saves them to the cloud layer for state persistency, which is asynchronous to event handling. When a user leaves a game, the cached state of user object will be saved to the cloud layer. The client layer provides

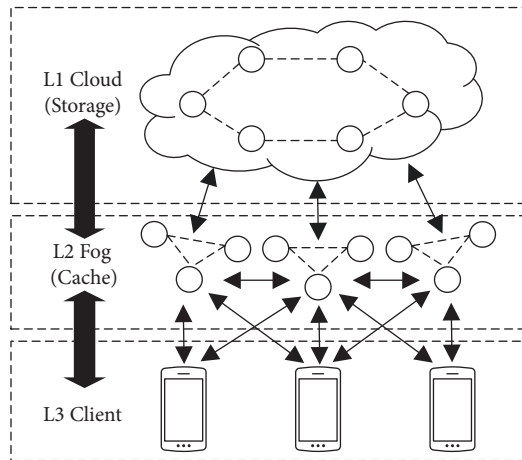


FIGURE 1: Virtual Net overall model.

user interfaces for receiving user operations and displaying updated states for user interaction. Virtual worlds are latency-sensitive applications. Yet, on one hand, clients require fast state update in user interaction [21]. On the other hand, the complexity of peer-to-peer routing slows down the process of content storage and retrieval [22]. Thus, the fog layer is padded between L1 and L3 to improve responsiveness in fault-tolerance.

The three-layer architecture is resilient. First, L1 and L2 can be individually scaled without affecting each other, since they are built for different purposes. The cloud layer focuses on the long-term storage of user content, which is only accessed at user login, logout, and periodic state checkpoint by the fog nodes. On the other hand, the fog layer maintains the latest state of user content and provides state recovery from intermittent client failure. Except for state initialization and checkpoint, L1 and L2 do not need to interact with each other. Besides, the model provides some extent of isolation of failure. The failure of one layer can be recovered by another layer, since each layer has a separate copy of content.

Different from the existing cloud-fog computing paradigm, computing resources in the cloud and fog layer are P2P nodes, like BitTorrent or eDonkey. Specifically, users contribute part of the computing resources from their devices which can be smartphones, laptops, desktop PCs, or even servers. A device is divided into one or several virtual nodes [23] for fine-grained load balancing. All virtual nodes are managed by a node pool. For different computing purposes, there are two types of virtual nodes: storage nodes and cache nodes. The storage nodes construct the cloud layer and the cache nodes construct the fog layer. Thus, Virtual Net is a decentralized computing paradigm. A client could be on the same device of a virtual node, like BitTorrent, or on a separate lightweight device.

3.1. P2P Cloud Layer. Object files are stored on the cloud layer through P2P file storage. Based on the file storage system, a content addressing scheme is devised, which can not only

provide flexibility in content identification and addressing but also provide integrity in object management.

3.1.1. File Storage. The TotalRecall [5] storage architecture is applied to manage the storage nodes for file storage. The details of the design and performance can be found in [5]. Here, only the overall mechanism is introduced. In TotalRecall, each node is assigned a unique hash code as the node ID. Also, each file has a file ID which is the hash checksum of the file. When a new file is created, the file is associated with a storage node, called the master node whose ID is closest to the file ID. Other nodes hosting the data of the file are called host nodes. Master nodes manage the location of host nodes and the version control for the associated files. Each storage node can be the master node for some files and the host node for other files. Thus, the entire storage node network forms a distributed hash table (DHT) for file lookup. To request a file, its master node is found first with the file ID. Then, based on the reply from the master node, the host nodes are located and the file can be retrieved (or reconstructed).

3.1.2. Content Addressing. To retrieve the objects from the cloud layer, object content needs to be identified and addressed. A hierarchical content addressing scheme is devised, which can facilitate content integrity check. The devised content addressing scheme has four hierarchies: inventory, objects, components, and files, as illustrated in Figure 2.

Inventory-level: each user has an inventory file, identified by the inventory ID which is the hash code of the user ID. An inventory contains all the object descriptions, consistently managing content identification and modification. Thus, to retrieve the object contents, the inventory file needs to be retrieved first. *Object-level:* an object is identified by the object hash code and composed of one or multiple components. *Component-level:* each component is identified by the component hash code. Object components are the categories of object resource files, which are classified into animation, sound, texture, script, etc. *File-level:* the actual files of objects are addressed by file IDs in object descriptions. Through files ID, the actual file can be retrieved either from the local cache or from the DHT of the cloud storage.

Based on the structure of the content addressing scheme, a Merkle tree [24] (Figure 2) can be hierarchically constructed with the file hash code, component hash code, object hash code, and inventory hash code. With the Merkle tree, the integrity of user content can be recursively checked and the number of hash comparisons can be largely reduced [25]. Typically, a client caches more than 500,000 files of user-created contents [26]. Thus, an exhaustive search of updated files will be inefficient. We conduct an experiment with 200 objects and more than 5,000 files. Compared with the file-level and object-level content integrity check [26], Figure 3 shows that the proposed four-level content integrity verification has fewer hash comparisons, especially with respect to a small number of file changes.

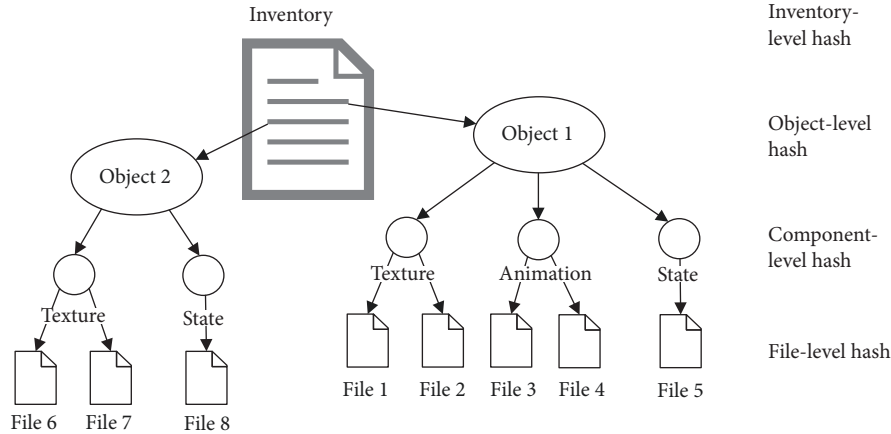


FIGURE 2: Illustration of the content addressing scheme hierarchy.

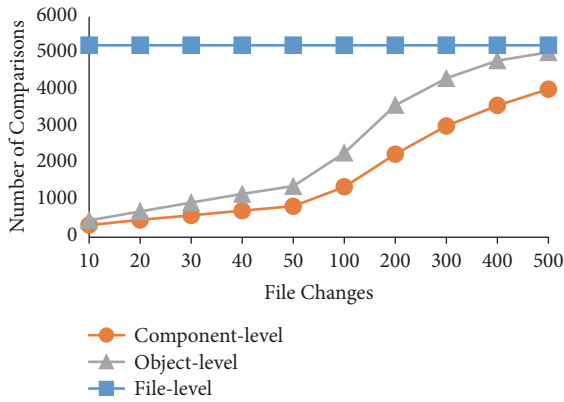


FIGURE 3: Number of hash comparisons with random file changes in content integrity check.

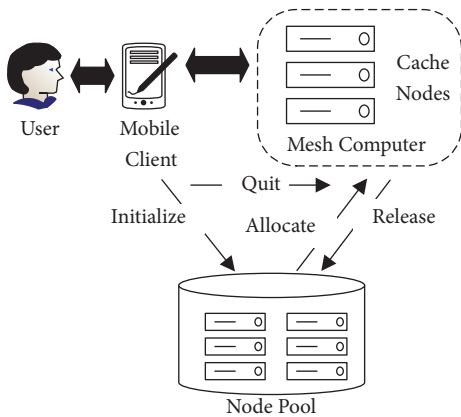


FIGURE 4: Mesh computer.

3.2. *P2P Fog Layer.* The fog layer is added between the cloud layer and the client layer to mask the latency of content storage and meanwhile provide fault-tolerance. From the user perspective, each user is allocated some cache nodes, when he/she is playing in a virtual world. These cache nodes provide the user some computing resources, forming a logical

computing unit. We call it mesh computer, as illustrated in Figure 4. When a user logs to the system, his/her client firstly initializes the mesh computer by requesting for some cache nodes from the node pool. The cache nodes then retrieve the content from the cloud layer. The client also retrieves the saved content from the cloud layer for content rendering and state synchronization. When the mesh computer receives a quit instruction from the client or the client is experiencing a long-term failure, the mesh computer will release the cache nodes to the node pool. Optimal resource allocation and cost minimization have been studied in [23], which is out of the scope of this paper.

Due to the unreliability characteristic of P2P nodes, they are subject to (either temporarily or permanently) failure. Thus, for reliability purpose, a mesh computer maintains multiple cache nodes which are the replicas of the same user content, called a replica group. Content will be transferred from failed nodes to live nodes. Replica group management has been studied in our previous work [27]. This paper focuses on replica state management in the following sections.

4. Object State Update

At the fog layer, it is important that all replicas of the same group maintain the same state of user objects so that any failure of a replica will not invalidate a user’s current state. The problem becomes challenging, since replicas could receive different sets of concurrent events from different senders and events could be received in different orders. The state machine replication (SMR) [28] approach is adopted to manage object state. SMR is a fault-tolerance model replicating a deterministic finite state machine on a set of distributed nodes, each of which has the same input, output, and state transfer. In an asynchronous cycle, firstly, clients send requests to all the nodes. On receiving the requests, a consensus protocol is triggered to determine the sequence of requests. Then, all nodes process the requests in the decided sequence so that they can reach the same new state. To reduce communication overhead, one replica is elected as the leader, coordinating membership reconfiguration and request ordering.

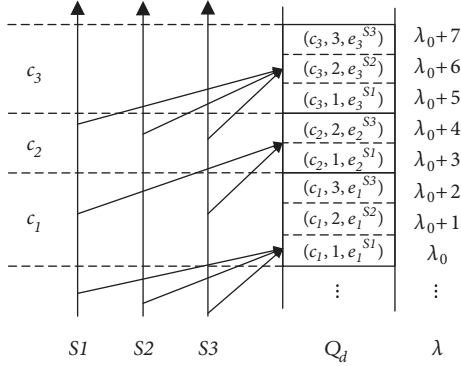


FIGURE 5: Relation of cycle (C_i), sender ID (S_i), event sequence number, delivery queue (Q_d), and delivery sequence (λ).

In virtual worlds, however, the consensus process adds a large delay in user interaction, because a requested event must be agreed on by all replicas after at least two communication rounds (i.e., four communication steps) [29] to reach an agreement, before it can be handled and replied to clients. The interaction delay issue in event handling is addressed based on the following observations. Due to users' limited perception range and motion speed, the number of event senders within a small period is fixed. Thus, the number of concurrent event senders within the period can be known a priori. Based on this observation, we propose a fast event delivery approach.

4.1. Fast Event Delivery. Fast event delivery allows a replica to directly deliver a received event through a cycle-event mapping, if it can ensure that the same event will eventually be delivered by all replicas. Specifically, the timeline is divided into infinite cycles of length Δt . From cycle c_0 , an event sender s periodically broadcasts an event to the replicas in each cycle. Each event is identified by the sender ID and the sequence number. The sequence number of the first event at cycle c_0 is 0. If there is no operation, s just broadcasts a no-op event. At the receiving end, c_0 is also known by all replicas. Each replica delivers an event with sequence number $c - c_0$ from s for cycle c , which is called the event of cycle c . Events for cycle c from different senders will be ordered by sender ID. If a replica does not receive the event for cycle c from s , it will start an instance of consensus for the cycle. In the consensus, if a replica has received the event for cycle c , that event will be decided by the leader and delivered by all replicas. Otherwise, they will decide and deliver an empty event for cycle c . Events will be delivered to a queue Q_d first and then sent to the application from the queue for handling in sequence.

The relation of cycle (c_i), sender ID, event sequence number, delivery queue (Q_d), and delivery sequence (λ) are illustrated in Figure 5. Specifically, in Q_d , the subscript of event e denotes the event sequence number which is equal to the event sending cycle. Thus, for the same cycle, the events in Q_d are sorted by the sender ID and mapped to the local index numbers in Q_d (i.e., the second member in the tuples

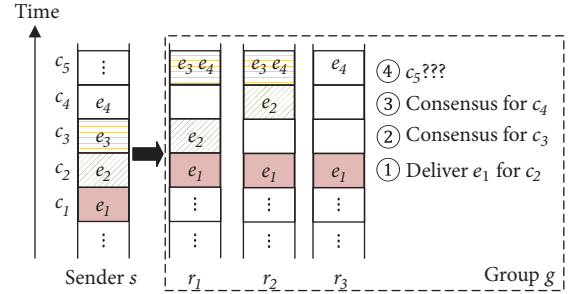


FIGURE 6: Illustration of fast event delivery.

of Q_d). λ represents the global index of events delivered to the application, which will be introduced in Section 4.1.4.

Figure 6 illustrates the fast event delivery process with one sender s and three replicas r_1, r_2 , and r_3 in the replica group g . s broadcasted four events, e_1, e_2, e_3 , and e_4 , at cycles c_1, c_2, c_3 , and c_4 to g . All replicas received e_1 at cycle c_2 and e_4 at c_5 . Only replica r_1 received e_2 at c_3 . No replica received e_3 at c_4 , but r_1 and r_2 received e_3 at c_5 . r_1, r_2 , and r_3 deliver e_1 for c_1 . They then collectively decide e_2 for c_3 and an empty event for c_4 through consensus. The first problem is how to decide c_5 and e_3 . According to the cycle-event mapping principle (i.e., one-cycle-one-event), the replicas should only deliver e_4 for c_5 and discard e_3 , leading to event loss. Before discussing the late events handling problem in detail, the settings and assumptions of the system will be introduced first.

4.1.1. Settings and Assumptions. For fault-tolerance, a replica group contains at least n replicas. The minimal group size n is determined by the content availability requirement [27] and the replica failure rate. To reduce replication overhead, each group also has an extra number e of nodes for lazy repair [5]. Once $e + 1$ replicas fail, new replicas will be added to recover the group size to $n + e$.

In each replica group, there is one non-replica node monitoring the state of all replicas, called Rendezvous [30]. A Rendezvous uses timeout to determine the state of replicas and then broadcasts their states to all replicas. Monitoring replica state is implemented by exchanging heartbeat messages between a replica and a Rendezvous. If the Rendezvous does not receive one heartbeat message within a cycle, the replica is treated as failed and removed from the group. New replicas are also added by the Rendezvous, once the group size is smaller than n . Rendezvous are reliable nodes, or called super-peers [19], since the existence of a group is determined by the Rendezvous. Once a Rendezvous fails, a new Rendezvous must be assigned to the replica group, which then rebuilds the replica group and recovers the object states from the cloud storage. By exchanging heartbeat messages, each replica learns the current membership of the group g , denoted by G , which contains all live replicas of group g . When the Rendezvous tells a replica that a member has failed or a new member is added, the replica will remove the member from G or add the member into G .

The system is assumed to be live. The SMR model contains three types of group-wide activities: leader election, group

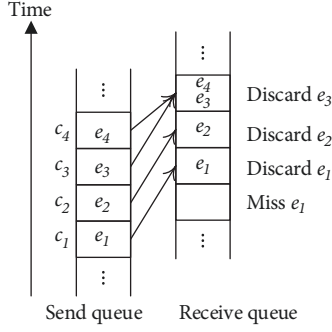


FIGURE 7: An extreme scenario of late event discard.

reconfiguration, and consensus. The liveness assumption ensures that when an activity is needed, it will eventually succeed after a finite number of failures.

Each replica group maintains a set of event senders. It is assumed that each event sender has an ID which is globally unique and sender IDs are comparable. Moreover, a replica group will append the join timestamp to sender IDs to distinguish a sender in two different joins of a sender set.

Let s be the ID of an event sender, c be a cycle number of a replica group g , and r_i be the i^{th} replica in group g . Some important relations of event, event sender, event sequence number, and cycle number are defined in Table 1. Other

notations used throughout the subsequent sections are listed in Table 1. Besides, event names are capitalized.

4.1.2. Late Event Handling. An event is late if the event of cycle c is received after c on all replicas. Formally, a late event e satisfies $Seq(e) = Seq(s, c) \wedge ReceiveCycle(e) > c$ on $r_i \in G$ and $Seq(e) = Seq(s, c) \wedge (ReceiveCycle(e) > c \vee ReceiveCycle(e) = \perp)$ on $r_j \in G \setminus \{r_i\}$. For example, e_3 in Figure 6 is a late event.

To ensure the agreement of cycle event delivery on all replicas, a late event can be simply discarded, since any event can be re-sent by a client with a new sequence number if the client does not receive the reply for the event for a period. However, if a sender's clock is temporarily out-of-sync with the replicas' clock or a large sending delay is experienced, a large number of events could be discarded and need to be re-sent, as shown in Figure 7.

To address the late event handling problem, a dynamic cycle event delivery approach is proposed, which includes two conditions for late event delivery. The purpose of the approach is to minimize the number of event discards, and meanwhile each replica can decide the delivery of late events with only local information. Below are the conditions of late event delivery. In short, only late and out-of-order events will be discarded.

- (1) At cycle c , all events from sender s with sequence number less than c will be deliverable. Formally, $\forall e(s, j_1), e(s, j_2), \dots, e(s, j_n)$,

$$\left. \begin{aligned} & Receive(e(s, j_1), c) \wedge \dots \wedge Receive(e(s, j_n), c) \\ & \wedge (DeliverCycle(e(s, j_1)) = \dots = DeliverCycle(e(s, j_n)) = \perp) \\ & \wedge (seq(s, j_1) < seq(s, c) \wedge \dots \wedge seq(s, j_n) < seq(s, c)) \end{aligned} \right\} \rightarrow Deliver(e(s, j_1), c) \wedge \dots \wedge Deliver(e(s, j_n), c). \quad (1)$$

- (2) At cycle c , an event will be nondeliverable, if one of its subsequent events has been delivered before c . Such event is a late and out-of-order event. Formally, $\forall e(s, j)$,

$$\left. \begin{aligned} & Receive(e(s, j), c) \\ & \wedge (\exists e(s, k) \wedge k > j) \\ & \wedge (DeliverCycle(e(s, j)) = \perp) \\ & \wedge (DeliverCycle(e(s, k)) < c) \end{aligned} \right\} \rightarrow \neg Deliver(e(s, j), c). \quad (2)$$

To implement the dynamic cycle event delivery approach, the lowest deliverable sequence number from any sender needs to be determined first for all cycles. Specifically, at cycle c , let $MinSeq(s, c)$ be the lowest sequence number of all undelivered events from sender s and $MinSeq(s, c) \leq seq(s, c)$. Also, let $MaxSeq(s, c)$ be the sequence number of the last delivered nonempty event in cycle c from s . Then, $MinSeq(s, c) = MaxSeq(s, c - 1) + 1$, where $MaxSeq(i, c - 1)$ is determined by the event delivery for cycle $c - 1$. Define the set of expected

deliverable events from s at cycle c by $\Omega(s, c) = [MinSeq(s, c), Seq(s, c)]$ and the set of actual received events $\Pi(s, c)$. The actual deliverable events from s at cycle c can be filtered by $\Omega(s, c) \cap \Pi(s, c)$, which excludes the late and out-of-order events.

4.1.3. Total-Order Event Delivery. Total-order event delivery is the key mechanism in object state update to ensure that all replicas in the same group can reach the same state along the same path of state transfer, if no more event is received. By applying the dynamic cycle event delivery approach, the event delivery for one cycle is described in Algorithm 1, where $E(c)$ stores the events from the consensus for cycle c and γ is the event delivery index in cycle c . γ is calculated by $\gamma = Seq(e) + \sum_{k=1}^{Index(Sender(e))-1} (Seq(k, c) + 1)$. (c, γ) and the calculation of γ ensure that the events in Q_d are sorted first by cycle number, then by sender index, and lastly by event sequence number, which can sort all events in the same order on all replicas.

In a run for cycle c , each replica firstly checks whether there are any events decided for the cycle from any consensus instance. If they exist, these events will be directly moved

TABLE 1: Notations.

Notations	Descriptions
Relations of event, event sequence number, sender, and cycle number	
$Seq(s, c)$	The sequence number of the event sent for cycle c from sender s
$Seq(e)$	The sequence number of event e
$e(s, j)$	The event of sequence number j sent from s
$Deliver(e, c)$	Return <i>true</i> if event e is deliverable at cycle c . Otherwise, return <i>false</i>
$DeliverCycle(e)$	Return the cycle in which event e is delivered. If e has not been delivered, return \perp .
$Receive(e, c)$	Return <i>true</i> if e has been received at cycle c . Otherwise, return <i>false</i> .
$ReceiveCycle(e)$	Return the cycle in which event e is received. If e has not been received, return \perp .
<i>Event of cycle</i>	The event sent by a sender for a cycle is called the event of the cycle. Given the first event sent from sender s at cycle c_0 . The event of cycle c satisfies $Seq(e) = Seq(s, c) = (c - c_0)$.
<i>Cycle open / close</i>	If a replica r_i has delivered the events for cycle c and moved to the next cycle $c + 1$, then c is closed and $c + 1$ is still open to r_i .
Other notations	
r_i	Replica i
r_L	Group leader
G	The set of group members
S	The set of event senders
U	Set of update recipients
g	Replica group g
s	Sender $s \in S$
$Index(s)$	The index of s in S sorted by sender ID
$Sender(e)$	The sender of event e
c	Cycle number c
Δt	Cycle length
$t_{start,s}$	The start time of the first cycle for s
$t_{send,s}(n)$	The time to send the n^{th} event from s
$t_{recv,s}(n)$	The time to receive the n^{th} event sent from s
t_{now}	Current time
$E / E(c)$	The set of decided events / decided events of c
Q_d	Delivery queue, containing the sorted events to be delivered to the application in sequence
(c, γ, e)	The event e delivered for cycle c with local index γ in Q_d
(λ, e)	The delivery sequence λ of event e in Q_d , one-to-one mapped onto (c, γ, e) for the given e .
$\Omega(s, c)$	The set of expected events which can be delivered to Q_d (called deliverable events) from s at cycle c
$\Pi(s, c)$	The set of actual received events from s at cycle c

to Q_d for event handling by the application. Otherwise, Algorithm 1 checks the condition $\Pi(s, c) = \Omega(s, c)$ for each sender s to ensure whether all expected deliverable events have been received. If there is an expected deliverable event not received in this cycle, then the replica will trigger a consensus instance to determine the event delivery for cycle c . Note that the cycle number c will only be increased if the events of the cycle have been delivered.

The proposed consensus algorithm is described in Algorithm 2 and illustrated in Figure 8. The consensus request is composed of the cycle number c and the sequence number

of all the expected deliverable events in c from all senders. By receiving the consensus request, each replica proposes the actual deliverable events to the leader. If a replica does not receive an expected event, it will propose \perp for the event. On receiving all proposals, the leader then decides the event for each sender and each expected sequence number. If at least one replica proposes a non- \perp and nonempty event $e(s, j)$ for $j \in \Omega(s, c)$, then $e(s, j)$ will be decided for sequence number j from s . Otherwise, an empty event will be decided for the slot. After the events of all slots have been decided, the leader will broadcast the decision to all replicas, and they will move the decided events to $E(c)$ for event delivery after receiving

```

1. For cycle  $c$ ,
2. If  $E(c) \neq \emptyset$ , then
3.    $Q_d \leftarrow Q_d \cup \{(c, \gamma, e(s, j)) \mid (c, e(s, j)) \in E(c)\}$ 
4.    $c \leftarrow c + 1$ 
5. Else,
6.   For  $\forall s \in S$ ,
7.     If  $\Pi(s, c) = \Omega(s, c)$ , then
8.        $Q_t \leftarrow Q_t \cup \{(c, \gamma, e(s, j)) \mid j \in \Omega(s, c)\}$ 
9.     Else,
10.       $Q_t \leftarrow \emptyset$ 
11.      Consensus for  $(c, \bigcup_{i=0}^{|\mathcal{S}|-1} \Omega(s, c))$ 
12.      End the loop
13. If  $Q_t \neq \emptyset$ , then
14.    $Q_d \leftarrow Q_d \cup Q_t$ 
15.    $c \leftarrow c + 1$ 

```

ALGORITHM 1: Event delivery.

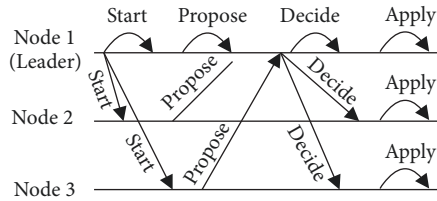


FIGURE 8: Illustration of the consensus protocol.

the decision. It is assumed that reliable point-to-point and multicast communication channels [29] are applied in the consensus protocol.

The complete algorithms of total-order event delivery, including event collection, event delivery, and consensus, can be found in Appendix A.

4.1.4. Garbage Collection. To avoid buffer overflow, events which have been delivered to the application need to be removed from Q_d to avoid Q_d from unlimited growth or even overflow. Due to asynchronous event handling, however, a replica r_i cannot safely remove a delivered event only with local information, because another replica r_j later may request the removed event for a given cycle if r_j does not receive it. Thus, determining the removable events is a challenging problem.

Learned from Algorithm 1, since all events in Q_d can be uniquely identified by (c, γ) , there exists a relation that maps each (c, γ) to a unique integer number $\lambda \in [0, \infty)$, called delivery sequence. Thus, events in Q_d can also be identified by (λ, e) and each (c, γ, e) and (λ, e) has a one-to-one mapping for the same e . Let Q_d be a sequence $e_i e_{i+1} \dots e_j \dots e_k$ mapped to $\lambda_i \lambda_{i+1} \dots \lambda_j \dots \lambda_k$. It can be observed that the events which can be safely pruned satisfy the following characteristics.

- (1) If e_j can be pruned from Q_d , then all events before e_j can all be pruned from Q_d .

- (2) Let $L = e_i e_{i+1} \dots e_j$ be the subsequence of Q_d . If L can be pruned from Q_d on one replica, then L can be pruned from Q_d of all the replicas in group g .

- (3) If e_j can be pruned from Q_d , then $\lambda_j \leq \lambda_c$, where λ_c is the sequence of the last event handled by the application, returned by the function $LastApplied(Q_d)$.

- (4) Trailing rounds with empty events cannot be removed from Q_d , since the events of these rounds may be used in consensus for deciding the value of late events.

(1) and (2) imply that there is a common latest applied event e_{cle} such that all the events delivered before e_{cle} (including e_{cle}) have been applied on all the replicas, whereas the events after e_{cle} are undecidable. The delivery sequence of e_{cle} is denoted by λ_{cle} . (3) implies that λ_{cle} cannot exceed λ_c . (4) restricts the range of garbage collection. Thus, all the events before λ_{cle} , which are not empty trailing events, can be safely removed from Q_d .

Based on the above observation, a gossip protocol is devised to learn λ_{cle} by exchanging λ_c of all replicas for safely estimating the earliest removable event, which is described in Algorithm 3. In the protocol, each replica periodically sends its λ_c to other replicas. A replica also caches the received λ_c . Based on the latest received λ_c from all replicas, λ_{cle} can be determined by the minimal λ_c . Then, λ_{cle} is adjusted to exclude the trailing empty events (Lines 12-16). Lastly, all the events before λ_{cle} are removed from Q_d . In Algorithm 3, Λ caches the received λ_c 's from all replicas in G . For a λ_c from r_i , if λ_c is greater than the cached value of r_i in Λ , the new λ_c can safely replace the existing one since λ_c from the same replica are monotonically nondecreasing.

4.1.5. Time Synchronization. In the fast event delivery approach, another key component is the synchronization of the start and end time of a cycle on event senders and recipients (all the replicas in group g) to minimize the chance of handling late events through consensus. Specifically, let Δt_n be the amount of network latency. Assume that the upper bound and lower bound of Δt_n , denoted by ΔT_n^L and ΔT_n^H ,

```

1. Given  $(c, \bigcup_{i=0}^{|\mathcal{S}|-1} \Omega(s, c))$ ,
2.  $r_i$  proposes:  $\{(s, j, e(s, j)) \mid s \in \mathcal{S} \wedge j \in \Omega(s, c) \cap \Pi(s, c)\}$ 
3.
4.  $r_L$  decides:
5.   For each  $s \in \mathcal{S}$  and  $j \in \Omega(s, c)$ ,
6.   // Proposals: the set of proposed events for  $c$  from all replicas;
7.   If  $\nexists e(s, j) \neq \perp \wedge e(s, j) \in \text{Proposals}$ , then
8.      $e(s, j) \leftarrow \text{Empty}$ 
9.      $D(c) \leftarrow D(c) \cup \{(c, e(s, j))\}$ 
10.
11.  $r_n$  applies the decision:  $E(c) \leftarrow D(c)$ 

```

ALGORITHM 2: Consensus.

```

1. On replica  $r_i$ :
2. Upon Timer TIMEOUT
3.    $\lambda_c \leftarrow \text{LastApplied}(Q_d)$ 
4.   Broadcast  $\lambda_c$  to all  $r \in G$ 
5.   Reset Timer
6.
7. On replica  $r_j$ :
8. Upon  $\lambda_c$  from  $r_i \wedge \lambda_c > \Lambda(r_i)$ 
9.    $\Lambda \leftarrow \Lambda \cup \{(r_i, \lambda_c)\}$ 
10.  If  $|\Lambda| \geq |G|$ , then
11.     $\lambda_{cle} \leftarrow \min\{\lambda_c \mid (r_i, \lambda_c) \in \Lambda\}$ 
12.    If  $(\lambda_{cle}, e_{cle}) = Q_d.\text{last}$  // last event in  $Q_d$ 
13.      Map  $\lambda_{cle}$  to  $(c, \gamma)$ 
14.      While  $\exists (c, \gamma, e) \in Q_d \wedge e = \text{Empty}$ 
15.         $\lambda_{cle} \leftarrow \lambda_{cle} - \{(c, \gamma, e) \mid (c, \gamma, e) \in Q_d\}$ 
16.         $c \leftarrow c - 1$ 
17.     $Q_d \leftarrow Q_d \setminus \{(\lambda, e) \mid \lambda \leq \lambda_{cle}\}$ 

```

ALGORITHM 3: Garbage collection protocol.

can be estimated such that most Δt_n falls within the range $[\Delta T_n^L, \Delta T_n^H]$. Then, cycle length Δt can be determined by $\Delta t = (\Delta T_n^H - \Delta T_n^L)$.

Let $t_{start,s}$ be the start time of the first cycle for event sender s designated by the replicas. Also, let $t_{send,s}(n)$ be the send time of the n^{th} event from s to the replicas. s firstly calculates $t_{send,s}(1)$ by $t_{send,s}(1) \leq t_{start,s} - \Delta T_n^L$. Then, in the n^{th} cycle, it calculates the sending time of the n^{th} event by $t_{send,s}(n) = t_{send,s}(1) + (n - 1) \cdot \Delta t$. At the receiving end, all replicas are timed to receive the n^{th} event from s at $t_{recv,s}(n) = t_{send,s}(n) + \Delta t = t_{start,s} + n \cdot \Delta t$. To timely collect received events, event collection and event delivery can be run by different threads with different buffers (see Appendix A for details).

A time server, such as a NTP server [31], can be deployed to the system for synchronizing the clock of event senders and recipients, which can improve the performance of the system.

4.2. Leader Election and Group Reconfiguration. Once the leader fails, a new leader is elected through leader election. Since both group reconfiguration and event handling rely on group leader, leader election has the highest priority in

the three routines. It interrupts any ongoing group reconfiguration or event delivery process. The leader election criterion is replica age, which increases by one after each group reconfiguration. Based on the assumption that node failure rate increase with time, the new leader is the youngest replica in the group.

Group reconfiguration adds new members for fault-tolerance. A group reconfiguration will be triggered once the group size is lower than n and recover group size to $n + e$. Group reconfiguration has higher priority than event delivery, so that new members can be quickly added to a group. After group reconfiguration, the leader will also notify all senders of the new configuration.

In both leader election and group reconfiguration, the leader will decide the current states, namely, Q_d , E , and G , and synchronize them to all replicas so that all replicas will load the same state after a leader election or a group reconfiguration, which is called state synchrony. For new replicas, the application state, the sequence of the last applied event λ_c , the time of the first cycle t_0 , the start time $t_{start,s}$ for each sender s , and the sender set \mathcal{S} are also synchronized from the leader for initialization. The detailed algorithms of leader election and group reconfiguration are in Appendix B.

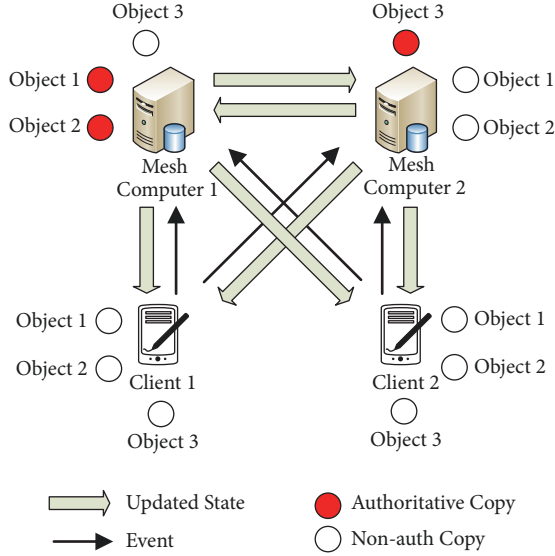


FIGURE 9: The flow of events and updated states in virtual world interaction.

5. Virtual World Interaction

Virtual world interaction describes how users manipulate the state of virtual objects and perceive the state change in a shared simulated environment. Following the definition, a virtual world interaction includes two steps. First, a user modifies the state of an object through operations. Second, the new object state is synchronized to other interested users. This section extends the proposed event delivery approach for supporting interactions in Virtual Net.

5.1. Flow of Events and Updates. An object is replicated on multiple hosts (i.e., clients and mesh computers) if multiple users operate the same object. To facilitate object state consistency management in interaction, the copies of objects are classified into authoritative copies and nonauthoritative copies. Each object can only have one authoritative copy but multiple nonauthoritative copies. An authoritative copy is maintained by one mesh computer, e.g., the object owner's. The nonauthoritative copies are maintained by the clients and the mesh computer of other interested users for fault-tolerance. Interest management has been intensively studied in [8] and thus is not discussed in this paper. It is only assumed that a user's interest scope is determined by his/her perception range in a virtual world, as illustrated in Figure 10.

By distinguishing authoritative copies from nonauthoritative copies, object state management is simplified to managing the state of an authoritative copy and synchronizing the updated state from the authoritative copy to nonauthoritative copies. For managing the authoritative copy, since the data is replicated to multiple nodes in a mesh computer, the fast event delivery approach is applied for maintaining the same state among these replicas.

From the perspective of an authoritative copy, an interaction includes receiving the event from one client, handling the event after it is delivered to the application, and multicasting

the updated state to all interested hosts. Figure 9 illustrates the flow of events and updated states. The events of an object are only sent from the clients to the mesh computer which maintains the authoritative copy of the object, while updated states are broadcast by the mesh computers to all the nonauthoritative copies. To support interaction, each mesh computer maintains two sets: the event sender set S and the update recipient set U .

5.2. Neighbors. To reduce overhead, each user only communicates with a limited number of peer users, called the neighbors. Due to user mobility, a user's neighbor may be frequently changed. A neighbor join happens when another user enters the perception range of a user. Likewise, a neighbor leave happens when a neighbor moves out of a user's perception range. For neighbour change, the key problem is to determine the same cycle of neighbor change on all replicas for their agreement on the cycle events. The join/leave cycle can be simply synchronized through consensus. However, high neighbor dynamics will increase the number consensus, resulting in high communication overhead and high interaction latency.

To apply the fast event delivery approach in neighbour change, the connectivity maintenance approaches of mutual notification [6] are employed. Specifically, two types of neighbour are introduced:

- (1) Perception neighbor set (N_p): the set of users and their virtual objects appearing in the perception range of a user.
- (2) Connectivity neighbor set (N_c): the set of users logically connected to the user.

Assume each user maintains a set of connectivity neighbors N_c . How to achieve it in a P2P virtual world can be found in [6]. A user (called *User i*) periodically exchanges its perception neighbor set N_p with the connectivity neighbors. Once a connectivity neighbor finds that another user should/should not be in N_p , it will notify *User i*. To facilitate description, some abstract functions are introduced:

- (i) *Multicast*(e, y, g): event e with sequence number y is sent to all replicas of group g .
- (ii) *Handle*(e): event e , which has been delivered to Q_d , is handled by the application.
- (iii) *Time*(t): set the timer to t , which will trigger a timeout event at t .
- (iv) *EVENT* $\leftarrow c$: assign content c to event *EVENT*.

5.3. Neighbor Join. Suppose *User j* is one of the connectivity neighbors of *User i*. *User j* discovers that another user *User k* is in the perception range but not in the N_p of *User i*; it will notify *User i* for adding the new neighbor with the following procedure. To distinguish clients from mesh computers, let p_i be the client of *User i* and r_i be the replica of group g_i (i.e., the mesh computer of *User i*), the same for p_j and p_k .

Step 1. p_j *Multicast*(*ADD_NEIGHBOR* $\leftarrow (p_k, G_k), y, g_i$).

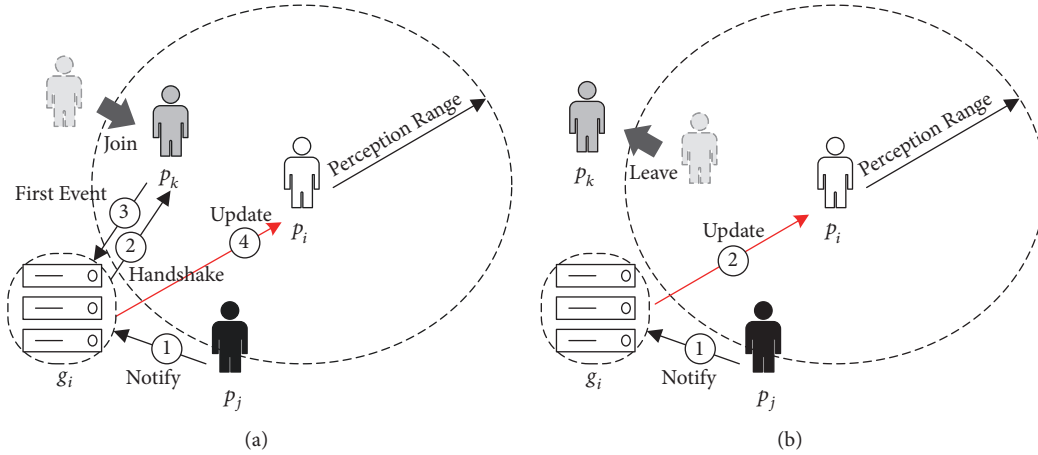


FIGURE 10: Neighbor change: (a) neighbor join; (b) neighbor leave.

Step 2. r_i *Handle*(ADD_NEIGHBOR) for cycle c , $\forall r_i \in G_i$.

- (a) r_i modifies $S \leftarrow S \cup \{p_k\}$, $U \leftarrow U \cup \{p_k, G_k\}$.
- (b) r_i calculates $t_{start,k} = t_{recv,j}(y) + n \cdot \Delta t$ and $t_{recv,k}(l) = t_{start,k} + \Delta t$.
- (c) r_i calculates cycle $c_k = \lceil (t_{recv,k}(l) - t_{now}) / \Delta t \rceil + c$.
- (d) r_i *Time*(c_k) for receiving the first event from p_k .
- (e) r_i sends (HANDSHAKE $\leftarrow (t_{start,k}, G_i)$) to p_k .

Step 3. p_k adds G_i to the recipient list.

Step 4. p_k Calculate $t_{send,k}(l) \leftarrow t_{start,k} - \Delta T_n^L$.

Step 5. p_k *Time*($t_{send,k}(l)$).

Step 6. p_k Multicast($e, 0, g_i$) at $t_{send,k}(l)$.

Step 7. p_k *Time*($t_{start,k} + \Delta t$) for the next event.

Step 8. r_i receives EVENT at c_k .

Step 9. r_i delivers EVENT for c_k .

The neighbor join process is illustrated in Figure 10(a). First, the connectivity neighbor p_j sends the ADD_NEIGHBOR event to all replicas in g_i for adding new neighbor p_k . Then, each replica r_i modifies the event sender and the update set recipient set, calculates the first event start time for event sending and receiving, and notifies the new neighbor p_k . The client p_k is timed to send the first event at $t_{recv,j}(y) + n \cdot \Delta t - \Delta T_n^L$ (where n , Δt , and ΔT_n^L are preconfigured). Meanwhile, the replicas of g_i are waiting for the event of the first cycle c_k at $t_{recv,k}$. Since the start time $t_{start,k}$ and the first event sequence number $j_0 = 0$ are known to both communication ends, they can individually calculate the time of subsequent event sending and receiving. At last, client p_i learns the new neighbor through the update from g_i and renders it to User i .

5.4. *Neighbor Leave*. The procedure of neighbor leave is similar to but simpler than the neighbor join procedure.

Suppose User j is one of the connectivity neighbors of User i . When c_i discovers that another user User k is out of the perception range but still in N_p of User i , it will notify User i with the following procedure.

Step 1. p_j Multicast(RM_NEIGHBOR $\leftarrow (k), y, g_i$).

Step 2. r_i *Handle*(RM_NEIGHBOR) for cycle c , $\forall r_i \in G_i$.

- (a) r_i modifies $S \leftarrow S \setminus \{p_k\}$, $U \leftarrow U \setminus \{p_k, G_k\}$ for cycle $c + 1$.

The neighbor leave process is illustrated in Figure 10(b). The connectivity neighbor p_j sends the RM_NEIGHBOR event to all replicas in g_i for removing the neighbor p_k , which can then remove p_k and G_k in both the event sender set and the event recipient set for cycle $c + 1$. Through the update from g_i , then client p_i learns the leave of neighbor p_k and removes p_k from display.

6. Theoretical Verification

The correctness of the state update design is determined by the state of all replicas in a group, as well as the clients. Firstly, the correctness of leader election and group reconfiguration are verified, since they support the other propositions. The proof of all lemmas and theorems can be found in Appendix C.

Lemma 1 (leader election synchrony). *All the live replicas in G maintain the same Q_d , E , and G after leader election.*

Lemma 2 (group reconfiguration synchrony). *All the live replicas in G maintain the same Q_d , E , and G after a group reconfiguration.*

Next, without loss of generality, the correctness of the consensus protocol is verified for an arbitrary cycle c . The validity property and the integrity property [29] are not verified here, since they are not related to the main result and

easy to be verified. Interested users can prove them. Here, only the agreement property is verified.

Lemma 3 (consensus agreement). *If a live replica $r_i \in G$ delivers an event e to $E(c)$ from a consensus instance for cycle c , then e is eventually delivered to $E(c)$ by all the live replicas.*

With the above lemmas, the main result can be obtained. But before it, an important property of the late event handling approach needs to be verified first.

Lemma 4 ($\Omega(s, c)$ agreement). *All the live replicas in G expect delivering the same set of events $\Omega(s, c)$ for sender $s \in S$ and cycle c .*

Now, the main result of theoretical verification can be presented with the following theorem and corollary.

Theorem 5 (total-order event delivery). *If a live replica $r_i \in G$ delivers two different events e_1 and e_2 into Q_d with λ_1 and λ_2 , then e_1 and e_2 will eventually be delivered into Q_d on all the live replicas with λ_1 and λ_2 being two non-negative integer numbers and $\lambda_1 \neq \lambda_2$.*

Corollary 6 (replica synchronization). *All the live replicas in G maintain the same state of their virtual objects.*

Another important result is the correctness of garbage collection, which is verified in Theorem 7.

Theorem 7 (garbage collection safety). *If event e is removed from Q_d on $r_i \in G$, then e has been handled by the application on all the live replicas in G .*

Based on Theorem 5, the correctness of the neighbor change procedures is shown with the following corollaries.

Corollary 8 (total-order event delivery with sender join). *All the live replicas in G deliver the same first event e_0 from a neighbor s with the same delivery sequence λ_0 .*

Corollary 9 (total-order event delivery with sender leave). *All the live replicas in G deliver the same last event e_∞ from a neighbor s with the same delivery sequence λ_∞ .*

7. Performance Analysis and Comparison

The performance of the proposed fast event delivery approach is studied in terms of synchronization delay and update loss rate. Three alternative approaches are introduced and compared with the proposed approach: the primary-backup approach, the reliable primary-backup approach, and the consensus-based total-order approach.

In the primary-backup approach [11], one replica is the primary replica and the rest are the backup replicas. The primary receives and handles all events and then broadcasts updates to recipients. Meanwhile, the primary replica sends the received events to backups for fault-tolerance. In a reliable primary-backup, the primary broadcasts the update only after the events have been reliably synchronized to all backups.

Note that the unreliable primary-backup approach does not ensure state consistency in case of primary failure. The consensus-based total-order approach [29] is similar to the proposed design, except that all events are delivered through consensus. Specifically, in each cycle, all replicas propose the received events within the cycle; the leader decides the events delivery order for the cycle.

Synchronization delay describes the time consumed in synchronizing the events over all live replicas. The primary-backup approach does not have a synchronization delay. In the reliable primary-backup approach, only 2 communication steps are involved in event synchronization: the primary broadcasts the events to all backups and collects the response from the backups. The consensus-based total-order approach needs one more communication step, as shown in Figure 8. In the proposed approach, synchronization delay is factored by the probability p_{sync} of triggering the consensus protocol.

Update loss rate describes the probability that a client does not receive the corresponding update after it sends an event to a mesh computer, due to event loss or update loss. In the primary-backup approach, update loss will occur as long as the channel between an event sender/update recipient and the primary replica is failed. In the consensus-based approach and the proposed approach, update loss occurs only when no replica receives the event or all replicas fail to send the update to a recipient. Moreover, assume that late and out-of-order events are discarded in all the approaches.

The performance comparison of different approaches is shown in Table 2, in which d_c denotes the delay in collecting a message from all replicas, d_m denotes the reliable multicast delay, and p_{loss} denotes the probability of message loss on a link.

The comparison result shows that if p_{sync} is small, i.e., transmission latency and clock offset are low, then the synchronization delay of the proposed approach is small and may even be close to that of the unreliable primary-backup approach. Thus the proposed approach can opportunistically provide higher responsiveness than the consensus-based total-order approach and the reliable primary-backup approach.

For update loss rate, $p_{loss}^n + (1 - p_{loss}^n)p_{loss}^n < p_{loss} + (1 - p_{loss})p_{loss}$ for $n \geq 2$, which can be proved as follows.

First, let $n = 2$. Then,

$$\begin{aligned} & p_{loss}^2 + (1 - p_{loss}^2)p_{loss}^2 - (p_{loss} + (1 - p_{loss})p_{loss}) \\ &= -p_{loss}(p_{loss} + 2)(p_{loss} - 1)^2 < 0. \end{aligned} \quad (3)$$

Thus, $p_{loss}^n + (1 - p_{loss}^n)p_{loss}^n < p_{loss} + (1 - p_{loss})p_{loss}$ for $n = 2$.

Second, let $f(x) = p_{loss}^x + (1 - p_{loss}^x)p_{loss}^x$, $x \in \mathbb{R}$. Then,

$$\begin{aligned} & \frac{\partial}{\partial x} (p_{loss}^x + (1 - p_{loss}^x)p_{loss}^x) \\ &= p_{loss}^x \ln(p_{loss}) + (1 - p_{loss}^x)p_{loss}^x \ln(1 - p_{loss}) \\ &+ (1 - p_{loss}^x)p_{loss}^x \ln(p_{loss}) < 0. \end{aligned} \quad (4)$$

Thus, $p_{loss}^n + (1 - p_{loss}^n)p_{loss}^n$ monotonically decreases along with n .

TABLE 2: Performance analysis and comparison results.

	Synchronization Delay	Update Loss Rate
Primary-backup	0	$p_{loss} + (1 - p_{loss})p_{loss}$
Reliable Primary-backup	$d_c + d_m$	$p_{loss} + (1 - p_{loss})p_{loss}$
Consensus-based	$d_c + 2d_m$	$p_{loss}^n + (1 - p_{loss}^n)p_{loss}^n$
Fast Event Delivery	$(d_c + 2d_m)p_{sync}$	$p_{loss}^n + (1 - p_{loss}^n)p_{loss}^n$

Therefore, $p_{loss}^n + (1 - p_{loss}^n)p_{loss}^n < p_{loss} + (1 - p_{loss})p_{loss}$ for $n \geq 2$. This shows that the consensus-based total-order approach and the proposed approach have lower update loss rate than the primary-back approaches.

8. Experiments and Results

8.1. Simulation Setup. The proposed model is evaluated by simulating distributed computing. Experiments are run in OMNeT++ to simulate message transmission in a network and event-based programming (simulation code at <https://github.com/sunnie1/VirtualNetEventHandling>). The simulation is run by sending events from 10 clients to a replica group, representing 10 neighbors. The replica group size is configured to 5. Cycle length is set to 200ms. In experiments, each client sends more than 9000 events to the replica group, which can simulate a half-hour game session with 200ms user operation interarrival time, applicable to most game genres [32]. After events are sorted and handled, updates will be transmitted to clients for collecting the statistic result. New replicas are generated by the Rendezvous; if the group size is lower than the availability threshold new replicas will be generated by the Rendezvous.

The network traffic model includes packet latency and packet loss. The packet loss rate is varied to simulate different rate of message loss rate p_{loss} . The packet delay is calculated by one-trip communication delay and network jitter. To facilitate simulation, network traffic is generated by a generating function from the analytical result of real data. Reference [33] suggests that the one-way delay between two hosts H_1 and H_2 can be modelled by $delay(H_1, H_2) = D_{min} + jitter$, where D_{min} is the minimum single-trip delay. $jitter$ is the network jitter caused by network congestion. In the experiments, D_{min} is configured to 50ms and $jitter$ is modelled by an exponential distribution and varied to simulate the variation of network latency.

To simulate replica failure, replica dynamics is characterized by session length which measures the length of time that a peer is continuously connected to a given P2P network, from its arrival to its departure [34]. Session length of P2P applications can be depicted by different stochastic models. Reference [34] shows that Weibull distribution or log-normal distribution fits the observation best. In this study, replica session length is modelled by a Weibull distribution with the mean value of half hour.

8.2. Experiment Results for Overall Performance. To verify the overall performance of event handling, three alternative event

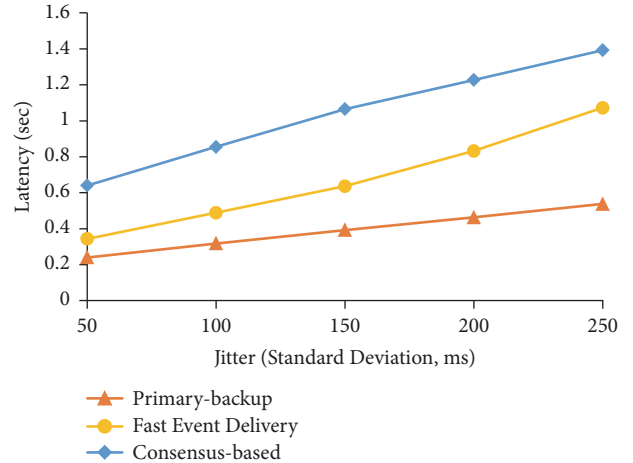


FIGURE 11: Interaction latency comparison in different models.

delivery approaches are implemented, including primary-backup, consensus-based total-order, and fast event delivery, for comparing their performances. Reliable primary-backup is not included in the comparison, since its performance is in-between the unreliable primary-backup approach and the consensus-based approach.

First, responsiveness is evaluated by comparing the interaction latencies in different approaches. Interaction latency includes both the round-trip end-to-end delay between an event sender and a group of replicas and the synchronization delay. The mean value of network jitter is fixed to 50ms, and the standard deviation is changed from 50ms to 250ms to simulate the scenario that events occasionally come late and out-of-order. The experiment result in Figure 11 shows that the fast event delivery approach provides much lower latency than the consensus-based total-order approach. Especially when the network latency is small, the responsiveness of the proposed approach is close to the primary-backup model. This is because the rate of triggering the consensus protocol decreases, when most events arrive before the end of cycles.

Second, end-to-end update delivery rate is evaluated by varying the message drop rate to simulate the change of p_{loss} from 0.3 to 0.7. In an asynchronous network, message loss cannot be distinguished from long message delay. Thus, update delivery timeout is used to cover both situations, which is configured to 5 seconds. The mean value of network jitter is fixed to 50ms to eliminate the interference of late events.

Figure 12 shows the update delivery rate of the three different approaches. Specifically, the update delivery rate of

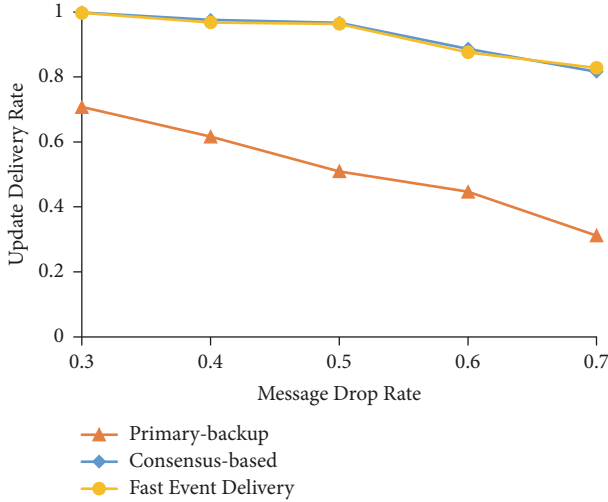


FIGURE 12: Update delivery rate with different message drop rate.

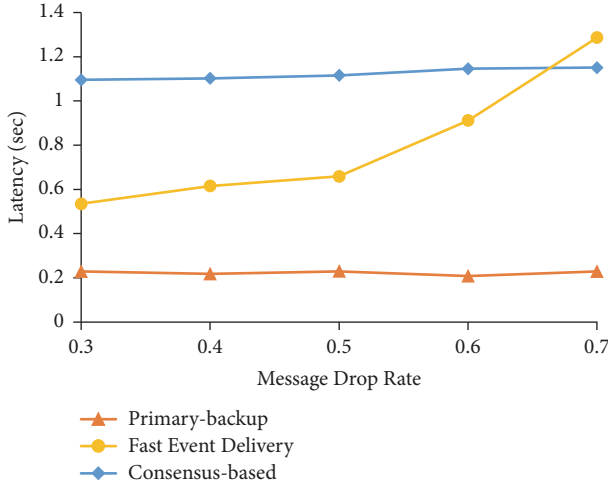


FIGURE 13: Interaction latency with different message drop rate.

the primary-backup approach is much lower than the other two approaches. Moreover, it drops quickly from around 0.7 to below 0.3 with the increase of p_{loss} , showing the rapid increase of update loss rate. In contrast, the update delivery rate of the consensus-based total-order approach and the proposed approach (overlapped) can remain high. Before p_{loss} is lower than 0.5, the update delivery rate of these two approaches is close to 1. When p_{loss} is lower than 0.5, the drop of the update delivery rate of them becomes evident. This is because, with the increase of message drop rate, more messages are received by none of the replicas.

In the same experiment, interaction latency is also studied with the change message drop rate. Figure 13 shows that, with the increase of message drop, a replica has a higher chance to miss the cycle events, such that $\Pi(s, c) \neq \Omega(s, c)$ and more consensus instances are triggered by replicas for event synchronization. This means that increasing message drop has a similar effect of increasing network jitter on the fast event delivery approach in interaction latency.

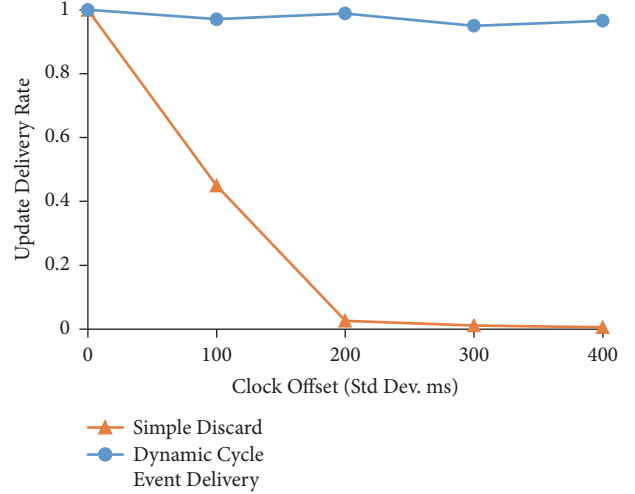


FIGURE 14: Update delivery rate with different late event handling approaches.

8.3. Experiment Results for Individual Improvements

8.3.1. Performance of Late Event Handling. The experiment result in this section verifies the late event handling approach. The proposed approach is compared with the simple discard approach which simply discards any late events. In the experiment, the timing of event sending is modified with clock error which is modelled with a normal distribution ($\mu = 0$). The standard deviation of the clock error is changed from 0 to 400ms to increase the rate of late event. Moreover, network jitter is fixed to 50ms and the message drop rate p_{loss} is fixed to 0 to eliminate their interference to the result.

Figure 14 shows that the proposed approach has a much higher update delivery rate than the simple discard approach. Especially when the clock error is higher than 300ms, simply discarding late events results in that almost no message is delivered, since most events will come late. On the other hand, through the proposed approach, the update delivery rate can be maintained as high as close to 1. But the update delivery rate is lower than 1, because a few late events do not meet the deliverability condition.

8.3.2. Performance of Garbage Collection. Garbage collection is tested to verify its effectiveness. The main purpose of the experiment is to show that the proposed mechanism can effectively limit the length of Q_d from overgrowth. Thus, the experiment is conducted with two different settings: one with garbage collection and the other without garbage collection. The increase of the delivery queue length (Q_d) is observed and compared for the two settings. Network jitter is fixed to 50ms and the message drop rate p_{loss} is configured to 0 to remove the interference of event loss in both settings, so that the length of Q_d is only determined by the number of events and garbage collection. In the second setting, the length of the garbage collection cycle is fixed to 5 seconds. The experiment result is shown in Figure 15. In the case that garbage collection is not applied, the length of Q_d quickly increases from several

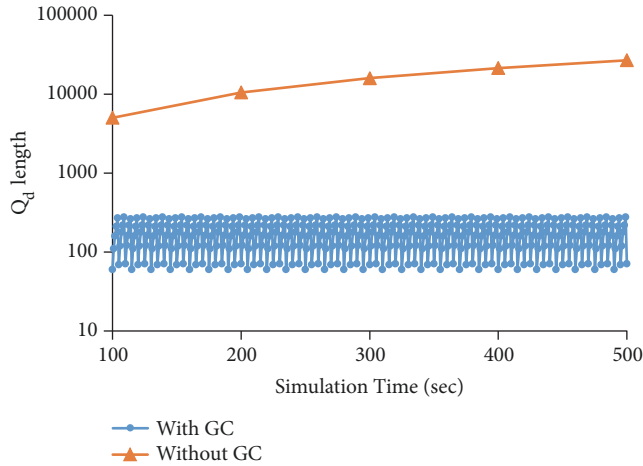


FIGURE 15: Delivery queue (Q_d) length along with simulation time with and without garbage collection (GC).

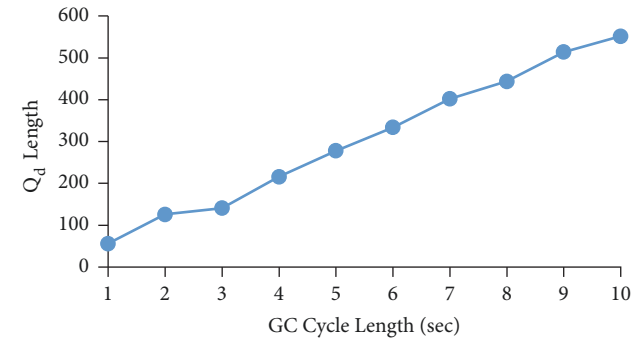


FIGURE 16: Trend of delivery queue (Q_d) length with different garbage collection (GC) cycle length.

thousand events to tens of thousands of events within 500 seconds. On the other hand, if garbage collection is applied, the change of Q_d is restrained within 300 hundred events. The comparison result shows that the proposed garbage collection protocol can effectively prevent Q_d from unlimited growth or even overflow.

Moreover, the cycle length of the gossip protocol is changed to show the control of the protocol on Q_d length. The experiment result is shown in Figure 16. When the cycle length of the gossip protocol increases from 1 second to 10 seconds, the length of Q_d changes from around 50 events to around 500 events. This experiment result shows that the length of Q_d is approximately linear to the cycle length of the gossip protocol. It implies that the length of Q_d can be effectively controlled by changing the cycle length. This control is useful because different virtual world applications could have different size of an event. If an application is required to cache large events, the length of Q_d will be reduced for the same space of event cache.

8.3.3. Performance of Time Synchronization. The performance improvement through time synchronization is shown in Figures 17 and 18. The main purpose of the experiment

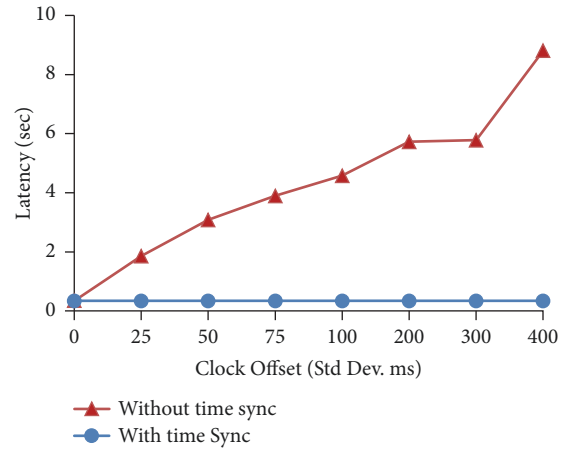


FIGURE 17: Interaction latency with different clock offset.

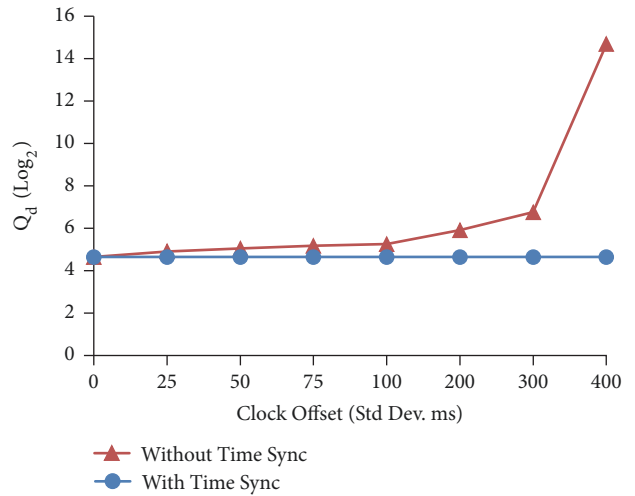


FIGURE 18: Delivery queue (Q_d) length with different clock offset.

is to show that the time synchronization mechanism has positive effect on interaction latency reduction and garbage collection. Thus, the experiment compares the performance of event handling in two different settings: one with time synchronization and the other without time synchronization. Clock error is added to the timing of event sending to simulate the scale of clock synchronization loss between the event senders and recipients. Network jitter is fixed to 50ms and the message drop rate p_{loss} is fixed to 0 to eliminate their interference to the result.

Figure 17 shows that if the clocks between event senders and event recipients are out-of-sync, there is an increase of interaction latency along with clock offset, because time synchronization loss increases the chance of triggering consensus for event delivery. Note that the interaction latency increase with clock offset is almost linear, clearly showing the impact of synchronization loss. In contrast, if time synchronization is applied before event transmission, the interaction latency is less than 1 second. This is because the number

consensus can be reduced and thus the communication steps for event delivery can be minimized accordingly.

Figure 18 shows the increase of the delivery queue (Q_d) also along with clock offset. Due to clock synchronization loss, more cycles are delivered with empty events, and thus more empty events are at the trail of Q_d . According to the rule of garbage collection, trailing rounds with empty events cannot be removed from Q_d . Thus, clock synchronization loss weakens the effectiveness of the garbage collection mechanism. Note that when the clock offset exceeds 300ms, the increase of Q_d will become large. In contrast, if time synchronization is applied, the length of Q_d does not increase along with clock offset.

8.4. Discussion. The experiment results for the overall performance show similar results to the theoretical analysis. Specifically, the proposed fast event delivery approach is reliable and can provide opportunistically high responsiveness, compared with the consensus-based total-order approach and the primary-backup approach, because this approach has the highest update delivery rate, and almost the same interaction latency as the primary-backup approach when the network latency is low. Thus, the overall performance of the proposed approach is better than the other two alternative approaches. The results also imply that, in practice, it is better to select cache nodes which are close to clients so that most cycle events can arrive on time at all replicas and they can be delivered without consensus. Then, interaction latency can be minimized.

The experiment results for the individual improvements show their effectiveness on system performance improvement. Specifically, the experiment result of late event handling shows that the proposed dynamic cycle event delivery approach can largely increase update delivery rate. High update delivery rate can reduce the chance of event resending which will lower down responsiveness in interaction and impact user experience. The experiment result of garbage collection shows its effectiveness in limiting the length of the event delivery queue. This is important, because it can not only avoid buffer overflow, but also restrict the time in traversing Q_d , if searching for a specific event is needed. Traversing a large buffer is slow and reduces system responsiveness. Lastly, the evaluation result of time synchronization shows its importance. Without the time synchronization between event senders and event recipients, the advantage of the fast event handling approach and the effect of the garbage collection mechanism diminishes.

9. Conclusions

With the popularity of mobile virtual worlds, scalability becomes an outstanding challenge in infrastructure development. The possibility of P2P technology is discussed to address the scalability problem. Different from existing P2P virtual worlds, client unreliability raises a new problem in mobile settings. This paper tries to solve the problem with a new hierarchical P2P computing model. Yet, rather than introducing every detail of the computing model, we focus

on object state update to avoid reinventing wheel. The core problem of state update is to maintain the replica state consistency without compromising system responsiveness. To address the problem, a fast event delivery approach is proposed. Based on this approach, we introduce the new virtual world interaction model to enable the interaction between multiple users.

Our work is important in providing a scalable infrastructure for mobile P2P virtual worlds. Based on the proposed Virtual Net architecture, there are some new research problems for building virtual world applications. First, our current approach still has the limitation in high responsiveness, since it belongs to the opportunistic category. To further improve system responsiveness without compromising state consistency, we plan to employ conflict-free replicated data types (CRDT) [35] to replace the consensus approach in event handling. With CRDT, events can be delivered in any sequence. However, events delivery in different sequences may cause user confusion with respect to continues events, such as avatar movement. Thus, it can be expected that the problem will be a combination of human-computer interaction (HCI) distributed computing. Moreover, the future study also includes the application and adaptation of cloud-fog computing techniques for contributed resource management, including cache node allocation, and P2P virtual world techniques to provide a complete and practical mobile P2P virtual world solution.

Appendix

A. Fast Event Delivery Protocols

The full set of the fast event delivery protocols are described in this appendix, which includes event collection, event delivery, and consensus. The payload of an event could be the operation of the event sender, *Empty*, or \perp (called \perp event). Note that if a reliable communication channel is required in a function, the keyword *Reliably* will be added before the send or broadcast operation. The implementation of a reliable channel can be found in [29]. Message names are capitalized and messages could contain some parameters. The notations used in the pseudocodes are listed in Table 3. Particularly, $G \cap R$ denotes the set of live replicas.

In each cycle Δt , the event collection protocol (Algorithm 4) periodically collects the received events from all senders in S in the receiving buffer Q , and to a temporary buffer Q_p . As described in the Late Event Handling section, not only the cycle events (i.e., $Seq(e) = Seq(i, c)$, Lines 8-11) but also the late and still deliverable events are collected (Lines 12-15). If any expected event is not received, a \perp event will be assigned to the corresponding event sequence number (Line 10). Likewise, a late event replaces an empty event with the same sequence number in Q_p (Lines 14-15). All undeliverable events will be discarded in event receiving (Line 3).

The event delivery protocol (Algorithm 5) is executed by detecting the condition satisfaction of a cycle. If the events of cycle $c - 1$ have been delivered and events of cycle c have either (1) been collected or (2) been decided from a consensus

TABLE 3: Notations in protocols.

Notations	Descriptions
r_i	Replica i
r_L	Group leader
r_c	Group leader candidate
G	The set of group members
R	The set of live replicas
S	The set of senders
$epoch$	The epoch number based on leader change
cid	Configuration ID
age	Age of a replica
c	Cycle number c
Δt	Cycle length
$E / E(c)$	The decided events / Decided events of c
Q_d	Delivery queue, containing the sorted events to be delivered to the application in sequence
Q_r	The received events
Q_p	The temporary buffer for event collection
s	Sender $s \in S$
$Index(s)$	The index of s in S sorted by sender ID
$Seq(e)$	The sequence number of event e
$Sender(e)$	The sender of event e
e	Event e , in format of $(Sender(e), Seq(e), Payload(e))$ where $Payload(e)$ denotes the payload of the event
$Seq(s, c)$	The event sequence number of c from s
$e(s, j)$	The event e of sequence number j from s .
$MinSeq(s, c)$	The lowest sequence number of all undelivered events from s in c
$MaxSeq(s, c)$	The sequence number of the last delivered non-empty event from s in c
$Consensus(c)$	Return <i>true</i> if there is a consensus on the fly for c
P	The set of cycles waiting consensus
Z	The set of cycles in consensus
LE	The flag of leader election
GE	The flag of group reconfiguration

1. On replica r_i :
2. Upon EVENT e
3. If $Seq(e) > MaxSeq(Leader(e), c - 1)$, then
4. $Q_r \leftarrow Q_r \cup \{e\}$
- 5.
6. Upon cycle TIMEOUT
7. $c \leftarrow c + 1$
8. For each $s \in S$,
9. If $e(s, Seq(s, c)) \notin Q_r$, then
10. $e \leftarrow (s, Seq(s, c), \perp)$
11. $Q_p \leftarrow Q_p \cup \{(c, e)\}$
12. For each $c' \in [0, c] \wedge s \in S \wedge e(s, c') \notin Q_d$, // Collect late and deliverable events
13. If $e(s, Seq(s, c')) \in Q_r$,
14. $Q_p \leftarrow Q_p \setminus \{e \mid e = (s, Seq(s, c'), \perp)\}$
15. $Q_p \leftarrow Q_p \cup \{(c, e(s, Seq(s, c')))\}$
16. $Q_r \leftarrow Q_r \setminus \{(e) \mid e(s, Seq(s, c)) \in Q_p\}$
17. Reset Timer cycle $\leftarrow \Delta t$

ALGORITHM 4: Event collection.

```

1. On replica  $r_i$ :
2. Upon  $(Q_d(c-1) \neq \emptyset) \wedge (E(c) \neq \emptyset \vee Q_p(c) \neq \emptyset) \wedge$ 
    $\neg \text{Consensus}(c)$ 
3.   If  $E(c) \neq \emptyset$ , then
4.      $D \leftarrow E(c)$ 
5.   Else
6.     For each  $s \in S \wedge j \in [\text{MinSeq}(s, c), \text{Seq}(s, c)]$ ,
7.        $c' \leftarrow c - (\text{Seq}(s, c) - j)$ 
8.        $T \leftarrow T \cup \{(c, e(s, c')) \mid (c', e(s, j)) \in Q_p\}$ 
9.     If  $\{(r, e) \mid \text{Payload}(e) = \perp\} \subseteq T$ , then
10.      QUERY  $\leftarrow (c, \bigcup_{s \in S} \text{MinSeq}(s, c), \bigcup_{s \in S} \text{Seq}(s, c))$ 
11.      Reliably send QUERY to  $r_L$ 
12.      End the procedure
13.   Else
14.      $D \leftarrow D \cup T$ 
15.   For each  $(c, e)$  in  $D$ ,
16.      $\gamma \leftarrow \text{Seq}(e) + \sum_{k=1}^{\text{Index}(\text{Sender}(e))-1} (\text{Seq}(k, c) + 1)$ 
17.      $Q_d \leftarrow Q_d \cup \{(c, \gamma, e)\}$ 
18.      $c \leftarrow c + 1$ 
19.
20. On leader  $r_L$ :
21. Upon QUERY  $(c, \bigcup_{s \in S} \text{MinSeq}(s, c), \bigcup_{s \in S} \text{Seq}(s, c))$  from  $r_i$ 
22.   For each  $s \in S \wedge j \in [\text{MinSeq}(s, c), \text{Seq}(s, c)]$ ,
23.      $c' \leftarrow c - (\text{Seq}(s, c) - j)$ 
24.      $R \leftarrow R \cup \{(c, e) \mid (c', e(i, j)) \in Q_p \vee (c, \gamma, e(s, j)) \in$ 
        $Q_d\}$ 
25.   If  $\{(c, e) \mid \text{Payload}(e) = \perp\} \subseteq R$ , then
26.     QUERY_REPLY  $\leftarrow (c, R)$ 
27.     Reliably send QUERY_REPLY to  $r_i$ 
28.   Else
29.      $P \leftarrow P \cup \{(c, \bigcup_{s \in S} \text{MinSeq}(s, c), \bigcup_{s \in S} \text{Seq}(s, c))\}$ 

```

ALGORITHM 5: Event delivery.

instance (Line 2), then the cycle c satisfies the condition of triggering the event delivery protocol. Thus, the execution of event delivery is asynchronous to event collection. For distributed agreement, the protocol firstly checks the second condition to ensure that the consensus result will be applied on all replicas. If the cycle is decided by a consensus instance, the decided events will be delivered no matter whether there is any nonempty event newly received for the cycle (Lines 3-4). Otherwise, events will be delivered from Q_p . If all expected events have been collected (Lines 6-8, 13-14), then they will be delivered to Q_d in the sequence of γ for cycle c . The range $[\text{MinSeq}(s, c), \text{Seq}(s, c)]$ specifies the deliverable sequence number for each sender s and cycle c . The calculation of γ (Line 16) ensures that all replicas can deliver the concurrent events from different senders in the same sequence. If there is any \perp event, a query message will be sent to the group leader. The set union $\bigcup_{s \in S} \text{MinSeq}(s, c)$ and $\bigcup_{s \in S} \text{Seq}(s, c)$ specify the lowest deliverable sequence number and the sequence number of the cycle respectively for all senders.

The leader checks locally the receipt of the events for the requested cycle in Q_p and Q_d . If all the expected events (for each sender s and each expected sequence number $[\text{MinSeq}(s, c), \text{Seq}(s, c)]$) of the requested cycle have been received, it will reply to them with the requesting replica (Lines 22-27).

Otherwise, the leader will initialize a new consensus instance for the cycle (Lines 28-29).

The consensus protocol (Algorithm 6) is run and instantiated for each requested cycle c . Note that the consensus protocol is only executed when there is no leader election (LE) or group reconfiguration (GE). Also, a message from a previous leader or a previous group configuration is not processed for consistency. Thus, these preconditions are added in all message handling procedures in the consensus protocol (Lines 8, 11, 19, and 26). First, the flags LE and GE are checked to ensure that the previous leader election and group reconfiguration have been finished. Then, the message sender's epoch and configuration ID are compared with the local epoch and configuration ID via adding the sender epoch and configuration ID to each message.

The consensus protocol is described in the Total-Order Event Delivery section in detail. A replica replies to the leader for the query of the events for cycle c only when the replica has passed the event collection of cycle c , which requires the following: (1) The decided events for cycle $c - 1$ have been delivered, if there is any. (2) The events collection for cycle c has been done. The leader will decide the events for c only after all proposals are received from all live replicas (Line 11). The Decide function determined the events of a given

```

1. On leader  $r_L$ :
2. Upon  $P \neq \emptyset \wedge CR = false \wedge LE = false$ 
3.   For each  $(c, \bigcup_{s \in S} MinSeq(s, c), \bigcup_{s \in S} Seq(s, c)) \in P \wedge c \notin Z$ ,
4.      $Z \leftarrow Z \cup \{c\}$ 
5.     QUERY  $\leftarrow (epoch, cid, c, \bigcup_{s \in S} MinSeq(s, c), \bigcup_{s \in S} Seq(s, c))$ 
6.     Reliably send QUERY to  $G \cap R$ 
7.
8. Upon QUERY_RESULT( $epoch', cid', r, W_i$ ) from  $r_i \wedge epoch' = epoch \wedge$ 
    $cid' = cid \wedge CR = false \wedge LE = false$ 
9.    $Q \leftarrow Q \cup \{(r_i, c, W_i)\}$ 
10.
11. Upon  $G \cap R \subseteq \{r_i \mid (n_i, c, W) \in Q\} \wedge CR = false \wedge LE = false$ 
12.    $R' \leftarrow Decide(c, Q)$ 
13.    $P \leftarrow P \setminus \{(c, \bigcup_{s \in S} MinSeq(s, c), \bigcup_{s \in S} Seq(s, c))\}$ 
14.    $Z \leftarrow Z \setminus \{c\}$ 
15.   DECISION  $\leftarrow (epoch, cid, c, W')$ 
16.   Reliably broadcast DECISION to  $G \cap R$ 
17.
18. On replica  $r_i$ :
19. Upon QUERY( $epoch', cid', r, \bigcup_{s \in S} MinSeq(s, c), \bigcup_{s \in S} Seq(s, c)$ ) from  $r_L$ 
    $\wedge epoch' = epoch \wedge cid' = cid \wedge GR = false \wedge LE = false$ 
20.   For each  $s \in S \wedge j \in [MinSeq(s, c), Seq(s, c)]$ ,
21.      $c' = c - (Seq(s, c) - j)$ 
22.      $W \leftarrow W \cup \{(c, e(s, j)) \mid (c', e(s, j)) \in Q_p \vee (c, \gamma, e(s, j)) \in Q_d\}$ 
23.     QUERY_RESULT  $\leftarrow (epoch, cid, c, W)$ 
24.     Reliably send QUERY_RESULT to  $r_L$ 
25.
26. Upon DECISION( $epoch, cid', c, W'$ ) from  $r_L \wedge epoch' = epoch \wedge cid' =$ 
    $cid \wedge GR = false \wedge LE = false$ 
27.    $E \leftarrow E \cup W'$ 
28.
29. Decide( $c, Q$ )
30.   For each  $s \in S$  and  $j \in [MinSeq(s, c), Seq(s, c)]$ ,
31.     If  $\exists e(s, j): e(s, j) \neq \perp \wedge e(s, j) \in \bigcup_{k \in G \cap R} W_k \wedge (r_k, c, W_k) \in Q$ , then
32.        $W' \leftarrow W' \cup \{(c, e(s, j))\}$ 
33.     Else
34.        $e \leftarrow (s, j, Empty)$ 
35.        $W' \leftarrow W' \cup \{(c, e)\}$ 
36.   Return  $W'$ 

```

ALGORITHM 6: Consensus.

cycle for each event sender (Lines 29-33) by the given set of received events for c from all replicas. For each sender s and event sequence number j , if all replicas propose \perp , then the payload of the event $e(s, j)$ will be decided with *Empty* (Lines 33-34). Otherwise, the event payload will be decided with the value of the proposal from any replica (Lines 30-31).

B. Leader Election and Group Reconfiguration Protocols

The notations that appeared in the leader election protocol and the group reconfiguration protocol follow the same convention listed in Table 3.

The leader election protocol (Algorithm 7) is triggered once the leader is not in the set of live replicas (Line 2). Each triggered replica checks whether it satisfies the condition

to be the candidate by calling the *SelectLeader* function (Lines 8-13). As described in the Leader Election and Group Reconfiguration section, the candidate has the smallest age. If multiple candidates have the same age, the one with the smallest ID is selected.

To achieve state synchrony, the candidate r_c sends the state query message (LE_QUERY) to all live replicas. If a replica has not learned the candidate or has learned a new candidate, the replica will reject the request from r_c by replying with the NACK message (Lines 17, 21-22). Otherwise, the replica will reply the state query with its state Q_d , E , epoch, and configuration (cid and G). r_c , on receiving the states from all live replicas, decides the latest consistent state (Lines 47-53) with the following functions.

- (i) *Longest*($\{Q_{d,i}\}$): selects the longest Q_d from all replicas.

```

1. On any replica:
2. Upon  $r_L \notin R \wedge r_c \neq self$ 
3.    $LE \leftarrow true$ 
4.    $r_c \leftarrow SelectLeader()$ 
5.   If  $r_c = self$  then
6.     Reliably broadcast LE_QUERY to  $G \cap R$ 
7.
8.    $SelectLeader()$ 
9.      $R_c := \{r_i \mid \forall r \in G \cap R, r_i.age \leq r.age\}$ 
10.    If  $|R_c| = 1$ , then
11.      Return  $r_i: r_i \in R_c$ 
12.    Else
13.      Return  $r_i: r_i \in R_c \wedge r_i.ID = \min\{r_j.ID \mid r_j \in R_c\}$ 
14.
15. On replica  $r_i$ :
16. Upon LE_QUERY from  $r_c$ 
17.   If  $r_c \in R \wedge r_c = SelectLeader()$ , then
18.      $LE \leftarrow true$ 
19.      $LE\_STATE \leftarrow (Q_d, E, cid, G, epoch)$ 
20.     Reliably send LAST_STATE to  $r_c$ 
21.   Else
22.     Reliably send NACK to  $r_c$ 
23.
24. Upon LOAD_LEADER( $Q_d', E', epoch', cid', G', Init$ )
    from  $r_c \wedge r_c \in R \wedge r_c = SelectLeader()$ 
25.    $(Q_d, E, cid, G, epoch) \leftarrow (Q_d', E', cid', G', epoch' + 1)$ 
26.    $(r_L, r_c) \leftarrow (r_c, \perp)$ 
27.    $LE \leftarrow false$ 
28.   If  $newReplica = true$ , then
29.     Initialize( $Init$ )
30.      $newReplica \leftarrow false$ 
31.
32. On leader candidate  $r_c$ :
33. Upon NACK from  $r_i$ 
34.   If  $self = Selectleader()$ 
35.     Reliably send LE_QUERY to  $r_i$ 
36.   Else
37.      $r_c \leftarrow \perp$ 
38.
39. Upon LE_STATE( $Q_{d,i}, E_i, cid_i, G_i, epoch_i$ ) from  $r_i$ 
40.    $Events \leftarrow Events \cup \{Q_{d,i}\}$ 
41.    $Decisions \leftarrow Decision \cup \{E_i\}$ 
42.    $Configs \leftarrow Configs \cup \{(cid_i, G_i)\}$ 
43.    $Epochs \leftarrow Epochs \cup \{epoch_i\}$ 
44.    $Senders \leftarrow Senders \cup \{r_i\}$ 
45.
46. Upon  $G \cap R \subseteq Senders$ 
47.    $Q_d \leftarrow Longest(Events)$ 
48.    $E \leftarrow Merge(Decisions)$ 
49.    $(cid, G) \leftarrow Latest(Configs)$ 
50.    $epoch \leftarrow Latest(Epochs)$ 
51.    $Init \leftarrow (t_0, \{t_{start,s} \mid s \in S\}, (\lambda_c, state), \{(r_i, r_i.age) \mid r_i \in G_T\}, S)$ 
52.    $LOAD\_LEADER \leftarrow (Q_d, E, epoch, cid, G, Init)$ 
53.   Reliably broadcast LOAD_LEADER to  $G \cap R$ 
54.   Broadcast  $G$  to  $S$ 

```

// New replica initialization is needed, in case that a group reconfiguration is interrupted by a leader election

// state: current application state

ALGORITHM 7: Leader election.

- (ii) $Merge(\{E_i\})$: returns the union of the Decision sets from all the replicas for all cycles.
- (iii) $Latest(\{cid_i, G_i\})$: returns the largest configuration ID cid and the corresponding replica set G , which represents the latest configuration seen by the group.
- (iv) $Latest(\{epoch_i\})$: returns the largest $epoch$ which represents the latest leader election seen by the group.

Moreover, in case of any unfinished group reconfiguration, additional states (including the time of the first cycle t_0 , the start time of the first event from all senders $\{t_{start,s} \mid s \in S\}$, the current state of the application and the corresponding delivered sequence of the event λ_c , the age of replicas, and the sender set (S)) are synchronized from r_c to new replicas for state initialization. After receiving the LE_STATE message from r_c , the replicas update their state to the decided value. Finally, all replicas load the r_c as the new leader and update the epoch by one.

The group reconfiguration protocol (Algorithm 8) is similar to the leader election protocol, except that it has lower priority, which is reflected by the precondition of checking the flag LE in all message handling procedures (Lines 11, 19, and 30). Group reconfiguration is triggered, when new replicas are added in the survival (i.e., $R \setminus G \neq \emptyset$). G_T caches the latest triggered reconfiguration to preclude any unnecessary retriggering (Lines 2-3). At the end of the reconfiguration, each replica updates the age of all replicas by one (Lines 26-27).

C. Proposition Proofs

See Lemma 1.

Proof. First, only one leader will eventually be elected by all the live replicas. It can be inferred by two cases. In the first case, the group is not partitioned. Then all the live replicas know each other, and the $SelectLeader$ function ensures that only one leader is elected by all live replicas. In the second case, the group is partitioned. Without loss of generality, suppose there are two different leaders, denoted by $r_{L,1}$ and $r_{L,2}$. $r_{L,1}$ is elected by replica set P and $r_{L,2}$ is elected by replica set Q . $r_{L,1} \notin Q$, $r_{L,2} \notin P$, and $P = G \setminus Q$. Following the partial synchrony assumption, if the replicas in P never know Q and vice versa, then either P or Q is removed by the Rendezvous of the group. Following the assumption that there is only one Rendezvous for each replica group, then only one partition, either P or Q , will eventually survive. Thus, eventually there is only one leader; either $r_{L,1}$ or $r_{L,2}$ is the leader of the group.

When a new leader is elected by all replicas, it will determine the Q_d , E , and G and broadcast them to all the live replicas. Through the reliable underlying channel, all replicas will eventually load the same Q_d , E , and G after leader election. Moreover, a monotonic epoch number is used to avoid a replica load state from an old leader. Thus, all live replicas will eventually load the same Q_d , E , and G after the leader election of the largest epoch. \square

See Lemma 2.

The proof of group reconfiguration synchrony is the same as that of leader election synchrony. Thus, it is not repeated here.

See Lemma 3.

Proof. If there is a leader election or a group reconfiguration before the consensus instance terminates, then Lemmas 1 and 2 ensure that all replicas will have e in $E(c)$. If there is no leader election or group reconfiguration before the consensus instance terminates, the reliable underlying communication channel ensures that all the live replicas will eventually receive the same decision from the leader. Since r_i has delivered e into $E(c)$, e is in the decision for cycle c . Therefore, e will be eventually received and delivered by all live replicas. \square

With the above lemmas, the main result can be obtained. But before it, an important property of the late event handling approach needs to be verified first.

See Lemma 4.

Proof. The lemma can be proved by induction.

Basis Step. When $c = c_0$, i.e., the cycle of receiving the first event from s based on $t_{recv,s}(l)$, then $\Omega(s, c) = \{e(s, 0)\}$.

Induction Step. Assume all the live replicas in G expect delivering the same set of events $\Omega(s, c_k)$ for sender $s \in S$ and cycle c_k ($c_k \geq c_0$). Then, for cycle $c_k + 1$, there are two cases for discussion.

- (1) If there is no consensus instance for cycle c_k , then $MaxSeq(s, c_k) = Seq(s, c_k)$ and $\Omega(s, c_k + 1) = \{e(s, c_k + 1)\}$ on all replicas.
- (2) If there is consensus instance for cycle c_k , then, following Lemma 3, all live replicas will eventually deliver the same events to $E(c_k)$. Let $Seq(s, j)$ be the maximal sequence number of nonempty events in $E(c_k)$. Then, $MaxSeq(s, c_k) = Seq(s, j)$ and $\Omega(s, c_k + 1) = \{e(s, j + 1), e(s, j + 2), \dots, e(s, c_{k+1})\}$ on all the live replicas.

By the principle of mathematical induction, it follows that the lemma is true for all cycles after c_0 . \square

See Theorem 5.

Proof. Since all replicas share the same sender set S , Lemmas 3 and 4 ensure that all the live replicas will eventually deliver the same set of events for any cycle, either directly from received events (Lines 5-14 of Algorithm 1) or from the consensus result (Lines 2-3 of Algorithm 1).

Let $e_1(s_1, j_1)$ and $e_2(s_2, j_2)$ be delivered on r for the cycle c_1 and cycle c_2 . If $c_1 = c_2 = c$, then (c, γ_1, e_1) and (c, γ_2, e_2) will be eventually delivered into Q_d of all replicas. If $c_1 \neq c_2$, then (c_1, γ_1, e_1) and (c_2, γ_2, e_2) will be eventually delivered into Q_d of all replicas. Moreover, since γ_1 and γ_2 are determined only by s_1, s_2, j_1 , and j_2 , $\gamma_1 \neq \gamma_2$ for different e_1 and e_2 . Since Q_d is linearly ordered by c and then by γ , there exists mapping from each unique (c, γ) to a unique nonnegative integer number λ and let $\varphi(c, \gamma) = \lambda$ be such mapping function. Let $\varphi(c_1, \gamma_1) = \lambda_1$ and $\varphi(c_2, \gamma_2) = \lambda_2$. Then, all replicas will eventually deliver (λ_1, e_1) and (λ_2, e_2) and $\lambda_1 \neq \lambda_2$. \square

<ol style="list-style-type: none"> 1. <u>On any replica:</u> 2. Upon $R \setminus G \neq \emptyset \wedge R \neq G_T \wedge LE = false$ 3. $G_T \leftarrow R$ 4. $GR \leftarrow true$ 5. If $r_L = self$, then 6. $cid \leftarrow cid + 1$ 7. $GR_QUERY \leftarrow (epoch, cid)$ 8. Reliably broadcast GR_QUERY to $G_T \cap R$ 9. 10. <u>On replica r_i:</u> 11. Upon $GR_QUERY(epoch', cid')$ from $r_L \wedge epoch' \geq epoch \wedge cid' > cid \wedge LE = false$ 12. $GR \leftarrow true$ 13. $cid \leftarrow cid'$ 14. If $epoch = 0$, then 15. $epoch \leftarrow epoch'$ 16. $GE_STATE \leftarrow (Q_d, E, cid, epoch)$ 17. Reliably send GE_STATE to r_L 18. 19. Upon $LOAD_CONFIG(Q_d', E', epoch', cid', G_T, Init)$ from $r_L \wedge epoch' = epoch \wedge cid' = cid \wedge LE = false$ 20. $(Q_d, E) \leftarrow (Q_d', E')$ 21. $G \leftarrow G_T$ 22. $LE \leftarrow false$ 23. If $newReplica = true$, then 24. $Initialize(Init)$ 25. $newReplica \leftarrow false$ 26. For each $r \in G \cap R$, 27. $r.age \leftarrow r.age + 1$ 28. 29. <u>On leader r_L:</u> 30. Upon $GE_STATE(Q_{d,i}, E_i, cid_i, epoch_i)$ from $r_i \wedge epoch_i = epoch \wedge cid_i = cid \wedge LE = false$ 31. $Events \leftarrow Events \cup \{Q_{d,i}\}$ 32. $Decisions \leftarrow Decision \cup \{E_i\}$ 33. $Senders \leftarrow Senders \cup \{r_i\}$ 34. 35. Upon $G_T \cap R \subseteq Senders$ 36. $Q_d \leftarrow Longest(Events)$ 37. $E \leftarrow Merge(Decisions)$ 38. $Init \leftarrow (t_0, \{t_{start,s} \mid s \in S\}, (\lambda_c, state), \{(r_i, r_i.age) \mid r_i \in G_T\}, S)$ 39. $LOAD_CONFIG \leftarrow (Q_d, E, epoch, cid, G_T, Init)$ 40. Reliably broadcast $LOAD_CONFIG$ to $G_T \cap R$ 41. Broadcast G_T to S 42. $G_T \leftarrow \emptyset$ 	<p>// Use cid to discard messages from a previous unfinished GR; // $epoch' \geq epoch$: for new members // $cid' > cid$: because the new cid has not been received</p> <p>// $state$: current application state</p>
--	--

ALGORITHM 8: Group reconfiguration.

See Corollary 6.

Corollary 6 can be directly inferred from Theorem 5.

See Theorem 7.

Proof. Theorem 5 ensures that if e is in Q_d of r_i , then e is or was in Q_d of all the live replicas in G with the same λ . In Algorithm 3, if e can be removed from r_i , then r_i must have received λ_c 's at least equal to λ from all the live replicas. Since events are delivered to the application in sequence, e must have been delivered to the application on all replicas. \square

See Corollary 8.

Proof. Theorem 5 ensures that the $ADD_NEIGHBOR$ event is delivered to Q_d of all replicas with the same λ . Since (λ, e) and (c, γ, e) have a one-to-one mapping for the same event, all replicas deliver $ADD_NEIGHBOR$ for the same cycle. Moreover, since $n, \Delta t$ are fixed, all replicas are timed to deliver the first event e_0 from s for the same future cycle c_k . Theorem 5 ensures that e_0 is delivered with the same delivery sequence λ_0 on all live replicas. \square

See Corollary 9.

Proof. Theorem 5 ensures that the RM_NEIGHBOR event is delivered to Q_d of all replicas with the same λ . Since (λ, e) and (c, γ, e) have a one-to-one mapping for the same event, all replicas handle RM_NEIGHBOR for the same cycle c . From cycle $c + 1$, s will be removed from S . Thus, all replicas will deliver the last event e_∞ of s at c . Theorem 5 ensures that e_∞ is delivered with the same delivery sequence λ_∞ on all live replicas. \square

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research is partially supported by the University of Macau Research Grant No. MYRG2017-00091-FST and MYRG2015-00043-FST.

References

- [1] Thinking Gaming. (2018, Jul.). "Top grossing iPhone - Games," <https://thinkgaming.com/app-sales-data/>.
- [2] Metro. (2018, Jul.). "Fortnite overtakes PUBG as the biggest video game in the world," <https://metro.co.uk/2018/02/09/fortnite-overtakes-pubg-biggest-video-game-world-7300442/>.
- [3] B. Knutsson, . Honghui Lu, . Wei Xu, and B. Hopkins, "Peer-to-peer support for massively multiplayer games," in *Proceedings of the IEEE INFOCOM 2004*, pp. 96–107, Hong Kong, PR China.
- [4] A. Yahyavi and B. Kemme, "Peer-to-peer architectures for massively multiplayer online games: A survey," *ACM Computing Surveys*, vol. 46, no. 1, 2013.
- [5] R. Bhagwan, K. Tati, Y. Cheng, S. Savage, and G. M. Voelker, "Total Recall: System Support for Automated Availability Management," in *In Proc. 1st Conf. Networked Systems Design and Implementation (NSDI)*, 2004.
- [6] E. Buyukkaya, M. Abdallah, and G. Simon, "A survey of peer-to-peer overlay approaches for networked virtual environments," *Peer-to-Peer Networking and Applications*, vol. 8, no. 2, pp. 276–300, 2013.
- [7] S.-Y. Hu, J.-F. Chen, and T.-H. Chen, "VON: A scalable peer-to-peer network for virtual environments," *IEEE Network*, vol. 20, no. 4, pp. 22–31, 2006.
- [8] T. Malherbe, *A Comparative study of interest management schemes in peer-to-peer massively multiuser networked virtual environment*, MEng. Thesis, Stellenbosch University, 2016, <http://hdl.handle.net/10019.1/100061>.
- [9] L. Ricci, L. Genovali, E. Carlini, and M. Coppola, "AOI-cast in distributed virtual environments: An approach based on delay tolerant reverse compass routing," *Concurrency Computation*, vol. 27, no. 9, pp. 2329–2350, 2015.
- [10] A. Yahyavi, K. Huguenin, J. Gascon-Samson, J. Kienzle, and B. Kemme, "Watchmen: Scalable Cheat-Resistant Support for Distributed Multi-player Online Games," in *Proceedings of the 2013 IEEE 33rd International Conference on Distributed Computing Systems (ICDCS)*, pp. 134–144, Philadelphia, PA, USA, July 2013.
- [11] H. A. Engelbrecht and J. S. Gilmore, "Pithos: Distributed storage for massive multi-user virtual environments," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 13, no. 3, 2017.
- [12] P. E. Ross, "Cloud computing's killer app: Gaming," *IEEE Spectrum*, vol. 46, no. 3, p. 14, 2009.
- [13] W. Cai, R. Shea, C.-Y. Huang et al., "A survey on cloud gaming: Future of computer games," *IEEE Access*, vol. 4, pp. 7605–7620, 2016.
- [14] S. Choy, B. Wong, G. Simon, and C. Rosenberg, "The brewing storm in cloud gaming: A measurement study on cloud to end-user latency," in *Proceedings of the 2012 11th Annual Workshop on Network and Systems Support for Games (NetGames)*, pp. 1–6, Venice, Italy, November 2012.
- [15] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.
- [16] M. R. Anawar, S. Wang, M. Azam Zia, A. K. Jadoon, U. Akram, and S. Raza, "Fog Computing: An Overview of Big IoT Data Analytics," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 7157192, 22 pages, 2018.
- [17] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.
- [18] E. C. P. Neto, G. Callou, and F. Aires, "An algorithm to optimise the load distribution of fog environments," in *Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017*, pp. 1292–1297, Canada, October 2017.
- [19] Y. Lin and H. Shen, "CloudFog: Leveraging Fog to Extend Cloud Gaming for Thin-Client MMOG with High Quality of Service," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 2, pp. 431–445, 2017.
- [20] N. Wang, B. Varghese, M. Matthaiou, and D. S. Nikolopoulos, "ENORM: a framework for edge node resource management," *IEEE Transactions on Services Computing*, 2017.
- [21] M. Claypool and K. Claypool, "Latency and player actions in online games," *Communications of the ACM*, vol. 49, no. 11, pp. 40–45, 2006.
- [22] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 2, pp. 72–93, 2005.
- [23] E. Carlini, L. Ricci, and M. Coppola, "Flexible load distribution for hybrid distributed virtual environments," *Future Generation Computer Systems*, vol. 29, no. 6, pp. 1561–1572, 2013.
- [24] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Advances in Cryptology — CRYPTO'87. CRYPTO 1987. Lecture Notes in Computer Science*, C. Pomerance, Ed., vol. 293, Springer, Berlin, Heidelberg, 1988.
- [25] C. C. Erway, A. K p c , C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," *ACM Transactions on Information and System Security*, vol. 17, no. 4, article 15, 2015.

- [26] C. Symborski, "Scalable user content distribution for massively multiplayer online worlds," *The Computer Journal*, vol. 41, no. 9, pp. 38–44, 2008.
- [27] B. Shen, J. Guo, and L. X. Li, "Cost optimization in persistent virtual world design," *Information Technology and Management*, vol. 19, no. 3, pp. 155–169, 2018.
- [28] F. B. Schneider, "Implementing fault-tolerant services using the state machine approach: a tutorial," *Computing Surveys*, vol. 22, no. 4, pp. 299–319, 1990.
- [29] C. Cachin, R. Guerraoui, and L. Rodrigues, *Introduction to Reliable and Secure Distributed Programming*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2nd edition, 2011.
- [30] A. Chandler and J. Finney, "Rendezvous: supporting real-time collaborative mobile gaming in high latency environments," in *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology (ACE'05)*. ACM, pp. 310–313, New York, NY, USA, June 2005.
- [31] D. L. Mills, "Internet time synchronization: the network time protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.
- [32] X. Che and B. Ip, "Packet-level traffic analysis of online games from the genre characteristics perspective," *Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 240–252, 2012.
- [33] S. Kaune, K. Pussep, C. Leng, A. Kovacevic, G. Tyson, and R. Steinmetz, "Modelling the internet delay space based on geographical locations," in *Proceedings of the 17th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP 2009*, pp. 301–310, Germany, February 2009.
- [34] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," in *Proceedings of the 6th ACM SIGCOMM on Internet Measurement Conference*, pp. 189–202, October 2006.
- [35] M. Shapiro, N. Pregoica, C. Baquero, and M. Zawirski, "Convergent and commutative replicated data types," *Bulletin of the European Association for Theoretical Computer Science. EATCS*, no. 104, pp. 67–88, 2011.