Mobile Phones as Ubiquitous Instruments: Towards Standardizing Performance Data on the Network

Nathan Bowen
Moorpark College
7075 Campus Road M142
Moorpark, CA 93021
nbowen@vcccd.edu

David Reeder
Mobile Sound
New York, NY
davidreeder@mobilesound.org

ABSTRACT

We propose a category of mobile phone musical instrument called 'ubiquitous instrument'. We present a conceptual framework that identifies ubiquitous instruments as a cultural site for popular and professional performance practice. Elements of this framework depend upon community involvement to standardize instrument profiles and inter-instrument communication shared across the network. We suggest OSC as the foundation for this network standard and propose definitions for the range and use of 'musical objects'. We present *4Quarters*, a collaborative music performance system driven by mobile phones, which approximates the ideal of ubiquitous instruments and demonstrates the use of musical objects.

1. INTRODUCTION

In recent years, there has been a steady increase in the number of mobile music applications. Hardware in mobile phones offer increasing computational power, improvements of onboard sensors, and greater storage capacity. Mobile phone software will steadily improve and operating systems will develop increasingly sophisticated means for gestural capture. Music apps, in particular, will reach a growing number of users as educational opportunities and institutional support provide economic incentives for designers of creative apps. We observe that mobile phones are already employed artistically with novel networking strategies and intentions. Mobile phone orchestras such as the Michigan Mobile Phone Ensemble and the Stanford Mobile Phone Orchestra generate an ever-growing repertoire of works. [1, 2, 3, 4] Meanwhile, there is a growing body of scholarship concerned with mobile phone music. [5, 6, 7]

Despite these trends, mobile phone music performance has not yet 'caught on' in popular consciousness. Many have not yet had the experience of *playing* a mobile phone. The mobile phone, as musical instrument, is still being defined and it will take time before a common performance practice crystallizes. [8]

At the same time, however, widespread acceptance of mobile phone technology, combined with new trends of wearable smart technology, indicate changes in our daily communications practice and our consumption of Internet media. Our dependence upon fixed, monolithic, desktop-based computing tools is quickly evolving towards the Internet of Things. This new world of ubiquitous computing distributes digital resources around us, embedding computing functionality into our mobile phones and the common objects we encounter.

Wide acceptance of mobile phones also gives authority to new user interfaces, patterns of network usage and daily interaction with global data sources. Features we now take for granted include malleable touch interfaces and the usual collection of onboard sensors such as gyroscopes, accelerometers, cameras, and microphones. [9]

1.1 Ideals and Purpose

Our shared digital environment is ready-made for the reoccurrence of social music-making. [6] Long ago, as in the time of Victorian England, musical instruments such as piano and guitar played a pivotal role in the social life of families and communities. Over time the role of these instruments was replaced by advances in recording and playback technology, beginning with the phonograph, later followed by home entertainment systems. Today, mobile phones have replaced the storage, selection, and playback functions of entertainment systems, while simultaneously introducing commonly understood, yet complex user interfaces. Meanwhile, the "social net" encourages the creation of mobile apps that unite groups of people over common interests. This combination of musical functionality, interfaces, and social engagement is reminiscent of the musical and social role played by the piano.

We believe this combination of trends enables an extraordinary future for instrument design, notably the creation of **ubiquitous instruments** in the digital realm and their capability to communicate across the network via **musical objects**.

1.2 Cultural Presence of Ubiquitous Instruments

We recognize ubiquitous instruments (UbIs) as a class of digital musical instruments (DMIs) that represent, as individual devices or in coordinated collections, the return of a well-known cultural site for music making. They are ubiquitous not just because they use mobile phones in their capacity as ubiquitous computing devices, but also because they provide a general set of performance interfaces that are natural to those using mobile phones on a daily basis. We are optimistic that these interfaces represent a flexibility of design that encourages expression

Copyright: © 2014 Nathan Bowen and David Reeder. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

from musicians at all levels, whether casual, amateur, professional or virtuoso, as do acoustic ubiquitous instruments of historical import such as piano or guitar. Section 2 provides our conceptual framework for the presence ubiquitous instruments. We believe this framework describes an historical situation that is repeating itself in the context of contemporary mobile phone technology.

It is not clear that a digital ubiquitous instrument has yet been invented. For those of us who aspire to the challenge, there are new problems to be solved. Mobile phone interfaces are more abstract than the piano: their visible construction does not imply a concrete structural mode of composition, as do the keys of a piano or the frets of a guitar. Problems posed by the non-concrete, malleable nature of contemporary digital devices are countered by at least two distinct advantages: 1) mobile phone networking allows UbIs to communicate, sharing performance data to influence sonic outcomes; 2) the language of sound elements (and associated compositional forms), employed by an UbI is flexible and can take any form.

1.3 Musical Objects

Allowing UbIs to share data over the network enables two or more instruments to interoperate, even when developed in isolation. Before we can rely on the integrity of shared network data, we must first standardize the set of **musical objects**, a consistent and predictable encyclopedia of identifiers designating typed values and functions in the service of musical expression.

The MIDI standard provides a well-known example of musical objects including definitions for velocity, aftertouch, pitch-shift, note on/off, system exclusive messages, and more. [11] These terms may still be valuable, though their transmission format and data types are bound by the requirements of legacy technology. We believe Open Sound Control (OSC) provides sufficient foundation for a public language of UbI expression. [12] Section 3 proposes that OSC be adapted to describe an extended set of musical objects.

1.4 Sonic Modalities

Mobile phones may offer sound constructions from any possible set of elemental sound units, or **sonic modalities**. We define a modality as a language of sound elements, such that each element is the smallest possible sound unit that clearly distinguishes itself against all other elements in the collection and which, in and of itself, contributes an essential and unique character to the whole collection, thereby defining the sonic space inhabited by the whole collection. A traditional example of this is a scale of pitches. We identify at least three sonic modalities: notes, patches, and samples. A modality of ‘notes’ indicates sounds organized in traditional sets of pitches (for example, as defined by Western European music theory). Clearly the tuning, scalar count, and distribution of notes depends upon the tradition from which they are drawn. ‘Samples’ indicate digitally sampled material, not whole works or tracks of extensive duration, but usually excerpts or smaller morphological divisions. Samples, like notes, may be further defined by the technique of production, in this case the character of the studio in which they were recorded or the practice of their reproduction (e.g. turntablism). By ‘patches’ we mean sounds generated by digital synthesis. We assume large sets of parameterized patches would be used to develop a naturally cohesive language of sounds. Use of real-time configurable patches assumes the presence of a synthesizer engine, whether in local software or available via the network. For an instrument to be sonically modal, we simply mean that it is capable of producing sounds from multiple, distinct sound languages. We believe many divisions between and within these initial three modalities are possible and should be encouraged by the needs and imagination of composers and instrument builders.

2. UBIQUITOUS INSTRUMENTS

The following conceptual framework for ubiquitous instruments is specific to the mobile phone (or other common mobile devices). Our approach reflects choices and assertions about instrument construction and the very definition of musical instruments in the context of mobile phones. We list a number of mutually dependent points, divided into three categories, the sum of which describe an environment supporting the cultural presence of ubiquitous instruments as a common cultural site for musical expression.

A full consideration of this list is beyond the scope of this paper. Some comments of clarification will follow.

2.1 Cultural and Economic Presence

i) There exists a compelling body of artistic output and achievement;
ii) There exists a steady repertoire of works, in a range of difficulties;
iii) Performance virtuosity is rewarded with social status and provides inspiration for anyone interested in the instrument;
iv) Educated listeners are intuitively able to distinguish between excellence versus gratuitous mistakes in performance, or superior form versus poor judgment within a composition;

2.2 Social and Personal Presence

v) Interested persons are able to understand the ubiquitous instrument interface immediately and intuitively, regardless of their degree of commitment to professional performance;
vi) Performance practice, at any level of expertise, plays a natural role in community building;
vii) Advancement of skill is cultivated and reinforced through institutional tradition and, sometimes, unforgiving discipline;

2.3 Structural and Compositional Presence

viii) Physical interfaces and musical complexity enable upward mobility towards mastery of virtuosic performance technique;

ix) Performers will have access to compositions in a variety of styles and genres;
x) UbIs will support one or more **sonic modalities** as a means to organize elemental units of sound into languages of sound;
xi) UbIs enable performance of works of any duration, from short to very long;
xii) Instrument interfaces (including orthogonality between interfaces) accommodate a wide and varying range of durations between which musical decisions may be made by the performer, whether kinetic or driven by meta data. This range of durations is bound by two extremes: the exploratory movement of casual players and the dexterity of virtuosos [13];
xiii) UbIs leverage characteristics unique to DMI including operating system services, data storage, and network capabilities;
xiv) UbIs send, receive, and execute **musical objects** shared between nodes (whether foreign or familiar) via instrument profiles and musical programming interfaces.

These points emphasize our sense of the gravity and rewards that come of building ubiquitous instruments. Whereas the first two categories (cultural and social) provide material that can be divided across individual development efforts, the points of structural and compositional presence, especially ***xiv***, can only be implemented by the larger community through dialog and shared results. We propose the use of OSC to define a public standard for shared musical objects in Section 3.

Points ***viii*** and ***xii*** consider the role of virtuosic performance technique. By comparison to their acoustic counterparts, digital UbIs generally have much less playable surface area. Consequently, the kinetic effort necessary to activate the instrument is diminished or simply different. However, we note that touch interfaces imply meta data. Unlike the GUI of personal computers, a mobile phone generally displays only one window at a time—one knows to look around the edges of the window, or behind the window, and knows that a phone represents a collection of active windows and settings, most of which are not visible. Meta thinking about windows, or malleable data planes, is a new shared construct [10] that can lead to new forms of virtuosity in mobile phone instrument performance.

The UbI abstraction implies that there will be a diverse class of digital instruments that embody and transform the ubiquitous qualities of the acoustic cultural instruments in our past. Not just one ubiquitous instrument will be created, but many – in the digital realm, singularities are replaced by multiplicities. Despite their variety, all UbIs should share basic interfaces and instrument profiles. Just as we naturally share images and URLs via social networks, we should share musical objects via protocols between instruments. It is most likely that the true UbI of popular consciousness will be a sum of the most accessible parts of all instruments, successful because they share network data and common musical practice in an intuitive and universal manner.

3. OSC AS A VEHICLE FOR MUSICAL OBJECTS

We believe OSC is capable of supporting a new language of musical objects, enabling data exchange and interoperability between ubiquitous instruments. The original architecture of OSC was neither intended as a standard, nor equipped for protocol negotiation. CNMAT core documents describe it as a “content format.” [13] Nevertheless, the existing foundation could be extended. We suggest adapting OSC in the following manner:

a) Standardize a set of public OSC Address Patterns for use in the global context;

b) Define OSC Packet exchanges in stateful protocols to support node registration and synchronization, and other uses of information sharing including network and routing configuration;

c) Establish conventions for when OSC Packets should be read as protocol packets (header plus payload) in a negotiated exchange;

d) Send OSC Packets via broadcast and multicast as well as point-to-point;

e) Enable OSC Methods and third-party APIs as data processing engines and/or data sources;

f) Establish conventions for negotiating OSC version, allowing graceful fallback to OSC version 1.0 in case of error.

Many of these uses are anticipated in the CNMAT core documents defining the scope and expected usage of OSC versions 1.0 and 1.1. [13, 14] We believe these additional features can be standardized without disrupting current conventions characterized by existing OSC applications.

Supporting this feature set requires updating the libraries that parse OSC Packets. Some suggest using colocated network services (e.g., Bonjour [15]) as a complement to services not normally provided by OSC. We believe it is simpler and more robust to bundle all network and information sharing functions together. We also expect to engage a period of protocol design and experimentation that requires full control over packet headers and data payloads. In the long run, a well-established protocol would ideally be integrated into the operating system of the supporting node and standardize its coexistence with supporting protocols.

3.1 Public, Vendor, and Private Address Patterns

To standardize the global namespace of OSC, we propose dividing the set of all possible OSC Address Patterns into three categories: public, vendor and private. Private OSC Addresses adhere to no global standard and are defined according to the needs of local systems that do not expect to interact with external systems. In this case, there are no limits on the namespace of OSC Addresses. Private usage is identical to the scope of OSC version 1.0 and is analogous to using Class A, IPv4 private 10.* addresses on a private IP network.

We suggest public and vendor addresses partition the global root of OSC Address Patterns into a small set of symbols, such as `/public` and `/vendor`, designating OSC Address trees with different functions. Public OSC Addresses would identify common resources available in the data ecosystem shared by all applications. Vendor OSC Addresses would include address roots assigned to public institutions such as universities, research organizations, and businesses. Though some subset of vendor OSC Addresses may be publicized for global use (and may eventually be promoted into the public tree), their owners would retain control over their address trees and would manage the release of changes on their own schedule. Public OSC Addresses, on the other hand, would be chosen only via deliberation of publicly elected interests driven by common consensus and would represent the most stable and reliable portion of the global OSC address space.

Section 4 of this paper provides an example of public OSC Address Patterns that the *4Quarters* application expects to send and receive between collaborating OSC implementations. It is natural that a subset of public addresses will terminate with OSC Methods named according to objects, functions, or data sources in the realm of mobile phone technology such as `gyroscope`, `accelerometer`, and other musical objects common to multi-touch, wearable, or ubiquitous interfaces.

We expect the public tree of OSC Addresses will preserve some or all of the MIDI standard under a single root, such as `/public/midi`. This root would represent a taxonomy of MIDI functions and data types. Where possible, these data types would be non-destructively expanded to match contemporary computing capabilities, such as encoding fields as floating point or 32-bit integer values. It may also be productive to update formats and ranges for channel naming, time codes, and note durations.

There are many other creative and computationally powerful means served by a public OSC Address space, including the definition of active processes that fetch data or manipulate other OSC Addresses. An example of this is the CNMAT `o.expr` library. [16] The functionality of such libraries might naturally reside in a modularized open-source stack for OSC parsing, capable of loading distinct modules for just-in-time evaluation. Such an infrastructure might also generalize to include plug-in modules that quickly represent common commandline applications or third-party data sources.

3.2 Universal Resource Identifier Scheme for OSC

As a means to unify this standardization effort, we suggest creating a new Universal Resource Identifier (URI) scheme for OSC, named `osc:` . [17] This form leads to many new questions beyond the scope of this paper. We know such a URI scheme is appropriate for many real-time uses of OSC which "fire and forget," sending a single OSC Message in real-time, encapsulated in a single OSC Bundle. Such uses map well to the Common Internet Scheme Syntax of RFC 1738 [18]:

```
osc://<host>:<port>/<OSC_Address_Pattern>
```

Perhaps an HTTP-like protocol supporting GET and POST will allow OSC clients to poll one another as an alternative to being passive receivers? Meanwhile, some solution is required to economize bandwidth by compressing repeated OSC Address prefixes or to use relative pathnames or URI fragments. [19, 20] Should state be maintained by the client or by the server offering a particular service? How might OSC Address Pattern matching be adapted in an OSC scheme?

Answering these questions cannot be decided in isolation. Developing a public standard requires organization amongst vested individuals and institutions, whether they are keen to develop ubiquitous instruments or simply interested to share OSC data in a reliable manner. We suggest that standardization of identifiers by a central authority follow the model of the Internet Corporation for Assigned Names and Numbers (http://icann.org) and development of technologies in the OSC ecosystem follow the model of the Internet Engineering Task Force (http://ietf.org). Should this effort clearly appeal to a critical mass of conference participants, the authors will endeavor to establish a well-publicized website and mailing lists as an open forum to spearhead an application to the Internet Assigned Numbers Authority (http://iana.org) for a provisional OSC URL scheme. [17]

4. CASE STUDY WITH 4QUARTERS

To illustrate our efforts to embody the UbI ideal, we present *4Quarters*. We view this instrument as a work in progress, not as a complete example of an ubiquitous instrument. [21] The final version may vary dramatically from the current example. *4Quarters* highlights elements of the UbI problem space (Section 2) and demonstrates the suggested promotion of OSC to a protocol with a global namespace (Section 3).

4.1 Architecture

4Quarters is modeled after Schnell and Battier's notion of a composed instrument system. [22] It is comprised of several mobile phones functioning as controllers and a desktop computer acting as server and audio production engine. Sound from all performers is funneled into a stereo image, allowing performances to take place easily in the home or in a concert hall, where the computer might be plugged into an entertainment system or concert PA. Communication between nodes requires a wireless local area network where OSC messages are sent to and from the server and between the mobile nodes (Figure 1).

The current implementation of *4Quarters* depends upon three core technologies: iOS, Java and Max/MSP. The server is built in Max/MSP augmented by a Java subsystem via the mxj object to handle OSC packets. Nodes are implemented as native iOS apps. All rely on a general purpose OSC parsing subsystem, derived from open source and enhanced per the suggestions in Section 3. We expect to release this customized OSC code to the public domain.

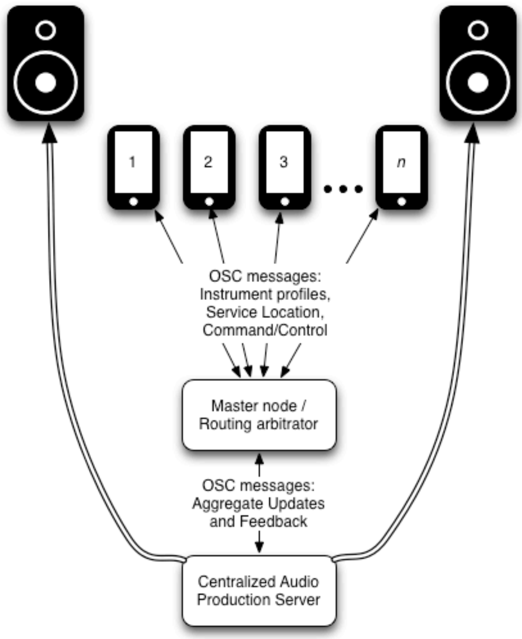

Figure 1. Basic architecture of *4Quarters*.

4.2 Performance Paradigm

As has been the intent in several mobile phone projects [23, 24, 25, 26], *4Quarters* is designed to be fundamentally collaborative, blurring the lines between composer, audience and performer. Presently the system can accommodate up to twelve players. A heads-up display from the server provides a central visual interface continuously updated with all user activity. Matching each user's assignment, the space is divided into four quadrants, each with its own color (Figure 2). If there are more than four players, various sound controls that correspond to single color may be divided amongst the players. Player color functions as a signifier for team coordination. For example, red player A may control volume, while red player B controls EQ.

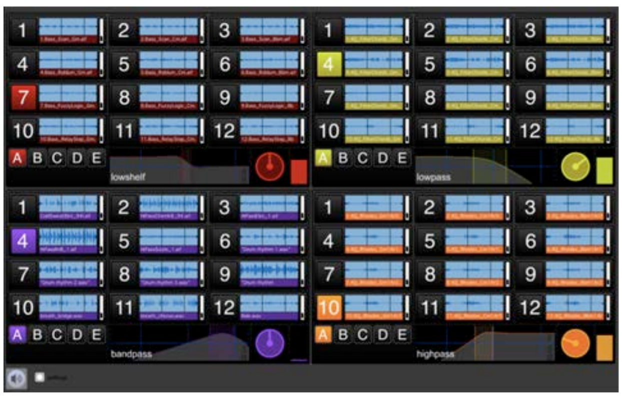

Figure 2. Master interface for all players in *4Quarters*.

4.3 Use of Sonic Modality

The original sonic modality of *4Quarters* is to sample and manipulate prerecorded audio files. This choice is a deliberate reaction to how portable music devices have replaced the record player as a music entertainment device. Generations of youth have exchanged music recordings via cassettes, compact discs, and MP3s as a form of social discourse and self-expression. Such exchanges contributed to the widespread cultural practice of remix and personalized playlists. Using samples as a sonic modality taps into popular music playback traditions.

Other modalities within *4Quarters* are under development: specifically, digital synthesis as a means to produce 'notes' and to manipulate sample data. Control parameters presently include sound file and sound bank selection, volume control, equalization, and envelopes. Each user may dynamically access up to sixty samples, divided into five banks of twelve samples each. Volume settings may be applied to individual samples or to the user's mix output.

4.4 Observations on Performance Practice

4Quarters assumes all participants to be in the same space engaged in a situated performance or jam session. This situation has ramifications that affect interface choices, particularly as it pertains to the visual interface. By testing this system with first-time users, several design problems have come to bear. For instance, eye contact between performers does not come without a conscientious effort. Eyes tend to focus on the central screen or on the device screens. As has been noted elsewhere [27, 28], facilitating eye contact between performers and/or the audience is crucial to help generate a sense of 'liveness.' Our next version of interface design minimizes the use of buttons and sliders. Essl and Rohs have authored an excellent 'orchestration' guide for present day mobile devices, highlighting various sensors and their expressive potential. [9] Through trial and error we believe a minimal phone interface coupled with a heavy reliance on tilt, gyroscope, accelerometer, and multi-touch will promote a performance experience that intuitively guides users to keep their heads up. Additionally, these particular sensors have the most potential for nuanced expression and to accommodate virtuosity.

4.5 OSC Implementation

Per Section 3, the *4Quarters* desktop OSC server and iPhone OSC clients communicate via a combination of broadcast advertisements and point-to-point messaging, not dissimilar from other implementations. [15, 29, 30] Advertisements have many functions depending upon their source and data payload, such as service location or asserting the current state of a node. It is assumed that advertisements are broadcast on the local network. In some cases, point-to-point exchanges are also broadcast – though OSC Packet headers will target a specific node, the packet payload may be of use to other nodes. This limits the need to re-broadcast or multicast ephemeral data to multiple endpoints and consumes no additional bandwidth.

Our node/server ecosystem consists of three entities: a single server for audio output, one or more performance nodes to trigger sounds on the server, and a master node to coordinate between the nodes and the server. The master node is elected from the set of performance nodes on a first-come, first-serve basis. OSC Packets are periodically broadcast as advertisements from each node as a means to identify their individual roles and capabilities. Each OSC receiver shares the same software subsystems and libraries to parse incoming and outgoing OSC Packets.

Every broadcast OSC message must lead with a bundle acting as a protocol packet header employing OSC Address Patterns as traditional packet data fields. At minimum these fields must include: protocol version, sender ID, receiver ID, packet type, and data. System level packet logic may evolve over time to indicate other services such as priority packet handling.

In summary, the packet ecosystem of *4Quarters* includes:

- service location advertisements
- node state updates
- broadcast state announcements
- point-to-point exchanges, especially for group join/departure
- data between performance nodes and the server

The first four packet types establish signaling and node awareness in a dynamic mesh network. The last type comprises the bulk of the packet transmissions and is where musical objects are exchanged (Figure 3).

4.6 Examples of Musical Objects

Lastly, we complement the discussion in Section 3 with simple, but concrete examples of OSC Address Patterns from the perspective of *4Quarters*. We suggest a possible hierarchy, absent OSC Arguments. In this case, we also omit details of syntax required by the proposed URI scheme. We defer to future work for a demonstration of full protocol exchanges containing full packet headers and data definitions describing network configuration and mobile phone performance vectors.

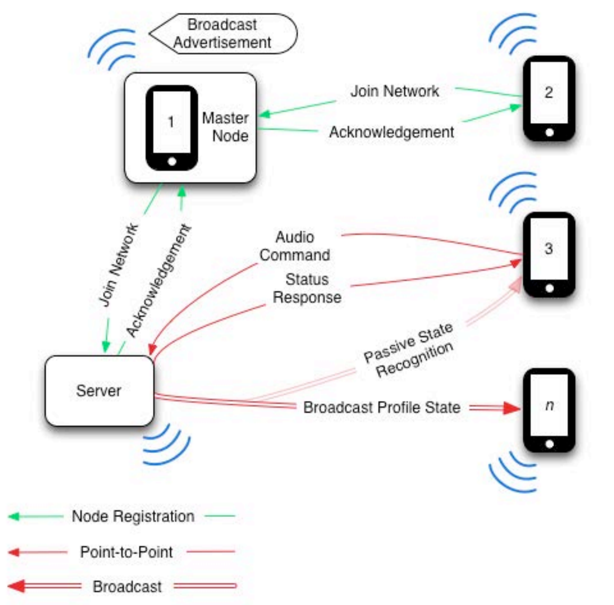

Figure 3. OSC Packet Ecosystem of *4Quarters*.

We assume the public portion of the protocol for musical objects contains definitions of common mobile phone sensors, notably `accelerometer` and `gyroscope`. Ellipses throughout indicate options for additional or extended OSC Address Patterns. This example includes an OSC pre-processing engine in the public portion, optimistically incorporating the `o.expr` framework [16] under `/public/library`. Vendor specific elements unique to *4Quarters* are listed in the private portion of the namespace under a root reserved for this project.

```
/osc/version
/osc/protocol

/public/sensor/accelerometer
/public/sensor/gyroscope
/public/sensor/. . .

/public/library/cnmat/o/expr
/public/library/ccm/. . .
/public/library/. . .

/vendor/4quarters/ui/slider/gain
/vendor/4quarters/ui/slider/Q
/vendor/4quarters/ui/xy
/vendor/4quarters/ui/. . .

/vendor/4quarters/settings/gain/enabled
/vendor/4quarters/settings/Q/enabled
/vendor/4quarters/settings/nodeIP
/vendor/4quarters/settings/serverIP
/vendor/4quarters/settings/. . .

/vendor/. . .
```

Note the distinction between the private project root `/vendor/4quarters`, over which we have full control, and the public OSC Methods under `/public/sensor` which we also implement, but do so against a public specification defining how those sensors should behave. Clearly, there would be other interests named under `/vendor` just as the hierarchy under `/public/sensor` and `/public/library` will develop according to needs demonstrated by the community. Just as each of these musical objects communicates a set of parameter values, each may be the root for any degree of complexity necessary to express its command set and internal state.

5. CONCLUSION

This paper presents definitions for **ubiquitous instruments**, **musical objects** and **sonic modalities** as a framework for discussion and development of new instruments on the mobile phone platform. We provide a conceptual framework for the re-emergence of socially shared musical performance. We call on the community to define a global standard of OSC exchanges for the purpose of enabling two or more distinct and separate mobile phone instruments to interoperate. *4Quarters* is presented as an incremental step towards the ideal of a ubiquitous instrument that also demonstrates how a standardized OSC protocol could be used.

We welcome ideas about other approaches to the ubiquitous instrument and its accompanying hurdles, such as using the public web infrastructure as an approximation to defining cultural sites for musical exchange. [30, 31]

6. REFERENCES

[1] G. Essl, "The Mobile Phone Ensemble as Classroom," in *Proceedings of the International Computer Music Conference (ICMC)*, Stony Brook, New York, 2010, pp. 506-509.

[2] Michigan Mobile Phone Ensemble. Accessed January 30, 2014, http://mopho.eecs.umich.edu/.

[3] J. Oh, J. Herrera, N. J. Bryan, L. Dahl, and G. Wang, "Evolving the Mobile Phone Orchestra," in *Proceedings of New Interfaces for Musical Expression (NIME),* Sydney, 2010, pp. 82-87.

[4] Stanford Mobile Phone Orchestra (MoPhO). Accessed January 30, 2014, http://mopho.stanford.edu/.

[5] F. Behrendt, *Mobile Sound. Media Art in Hybrid Spaces.* Ph.D. Thesis, University of Sussex, 2010.

[6] N. Bowen, *Mobile Phones, Group Improvisation, and Music: Trends in Digital Socialized Music Making*. Ph.D. Dissertation, Graduate Center, City University of New York (CUNY), 2013.

[7] L. Gaye, L.E. Holmquist, F. Behrendt, and A. Tanaka, "Mobile Music Technology: Report on an Emerging Community," in *Proceedings of New Inter-faces for Musical Expression (NIME)*, Paris, 2006, pp. 22-25.

[8] C. Brown, J. Bischoff, and T. Perkis, "Bringing Digital Music to Life," *Computer Music Journal,* vol. 20, no. 2, pp. 28–32, 1996.

[9] G. Essl and M. Rohs, "Interactivity for Mobile Music-Making," *Organised Sound,* vol. 14, no. 2, pp. 197-207, 2009.

[10] A. Tanaka, "Mapping Out Instruments, Affordances, and Mobiles," in *Proceedings of New Interfaces for Musical Expression (NIME),* Sydney, 2010, pp. 88-93.

[11] MIDI.org. Accessed January 30, 2014, http://www.midi.org/techspecs.

[12] M. Wright, "Open Sound Control: an enabling technology for musical networking," *Organised Sound*, vol. 10, no. 3, pp. 193-200, 2005.

[13] A. Schmeder, A. Freed, and D. Wessel, "Best Practices for Open Sound Control," in *Proceedings of the Linux Audio Conference (LAC),* Utrecht, NL, 2010, pp. 131-140.

[14] A. Schmeder and A. Freed, "Features and Future of Open Sound Control version 1.1 for NIME," in *Proceedings of the Conference on New Interfaces for Musical Expression (NIME)*, Pittsburgh, PA, 2009, pp. 116-120.

[15] G. Essl, "Automated Ad Hoc Networking for Mobile and Hybrid Music Performance," in *Proceedings of the International Computer Music Conference (ICMC),* Huddersfield, 2011, pp. 399-402.

[16] J. MacCallum, A. Freed, and D. Wessel, "Agile Interface Development using OSC Expressions and Process Migration," in *Proceedings of the Conference on New Interfaces for Musical Expression (NIME)*, Daejeon, Korea, 2013, pp. 347-351.

[17] Internet Engineering Task Force. Guidelines and Registration Procedures for New URI Schemes. Accessed March 31, 2014, https://www.ietf.org/rfc/rfc4395.txt.

[18] Internet Engineering Task Force. Uniform Resource Locators (URL). Accessed March 31, 2014, https://www.ietf.org/rfc/rfc1738.txt.

[19] J. Kleimola and P.J. Mcglynn, "Improving the Efficiency of Open Sound Control with Compressed Address Strings," in *Proceedings of the 8th Sound and Music Computing Conference (SMC)*, Padova, Italy, 2011.

[20] Internet Engineering Task Force. Uniform Resource Identifier (URI): Generic Syntax. Accessed March 31, 2014, https://www.ietf.org/rfc/rfc3986.txt.

[21] N. Bowen, "4Quarters: Real-time Collaborative Music Environment for Mobile Phones," in *Proceedings of the 1st Symposium on Laptop Ensembles & Orchestras (SLEO),* Baton Rouge, LA, 2012, pp. 83-87.

[22] N. Schnell and M. Battier, "Introducing composed instruments, technical and musicological implications," in *Proceedings of New Interfaces for Musical Expression (NIME),* Dublin, Ireland, 2002, pp. 156-160.

[23] Ligna, "Dial the signals! Radio Concert for 144 Mobile Phones." Posted July, 2009, accessed January 30, 2014, http://ligna.blogspot.com/2009/07/dial-signals-radio-concert-for-144.html.

[24] SAME Project, 2007-2010. Accessed January 30, 2014, http://www.sameproject.eu/.

[25] A. Tanaka, "Mobile Music-making," in *Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME),* Hamamatsu, Japan, pp. 154-156.

[26] J. Thomson and A. Craighead, *Telephony*. Thomson & Craighead, 2001, accessed January 30, 2014, http://www.thomson-craighead.net/docs/teleph.html.

[27] M. Blain, "Issue in Instrumental Design: the Ontological Problem (Opportunity?) of 'Liveness' for a Laptop Ensemble," in *Proceedings of the 1st Symposium on Laptop Ensembles & Orchestras (SLEO),* Baton Rouge, LA, 2012, pp. 63-71.

[28] T. Edwards and R.B. Sutherland, "Eyes Off the Screen! Techniques for Restoring Visual Freedom in LEO Performance," in *Proceedings of the 1st Symposium on Laptop Ensembles & Orchestras (SLEO),* Baton Rouge, LA, 2012, pp. 33-40.

[29] J. Allison, Y. Oh, and B. Taylor, "NEXUS: Collaborative Performance for the Masses, Handling Instrument Interface Distribution through the Web," in *Proceedings of the Conference on New Interfaces for Musical Expression (NIME)*, Daejeon, Korea, 2013, pp. 1-6.

[30] K. Umezaki and C. Lavender, "The MINC Player Management Project," in *Proceedings of the 1st Symposium on Laptop Ensembles & Orchestras (SLEO),* Baton Rouge, LA, 2012, p. 134.

[31] C. Roberts, G. Wakefield, and M. Wright, "The Web Browser As Synthesizer and Interface," in *Proceedings of New Interfaces for Musical Expression (NIME)*, Daejeon, Korea, 2013, pp. 313-318.