Requirements for a file format
for smart musical instruments

Luca Turchet
Center for Digital Music
Queen Mary University of London
London, UK
Department of Information Engineering and Computer Science
University of Trento
Trento, Italy
luca.turchet@unitn.it

Panos Kudumakis
Center for Digital Music
Queen Mary University of London
London, UK
p.kudumakis@qmul.ac.uk

Abstract—Smart musical instruments are an emerging category of musical instruments characterized by sensors, actuators, wireless connectivity, and embedded intelligence. To date, a topic that has received remarkably little attention in smart musical instruments research is that of defining an interoperable file format for the exchange of content produced by this class of instruments. In this paper we preliminary investigate the design of a format specific to smart musical instruments but that at the same time enables interoperability with other devices. We adopted a participatory design methodology consisting of a set of interviews with studio producers. The purpose of such interviews was that of identifying a set of use cases for a format encoding data generated by smart musical instruments, with the end goal of gathering requirements for its design.

Index Terms—Smart musical instruments, Internet of Musical Things, interoperability

I. INTRODUCTION

The "smart musical instruments" are an emerging family of digital musical instruments [1], which draws upon different lines of existing research including augmented instruments [2], embedded acoustic instruments, [3], and networked music performance systems [4]. At hardware level, this category of instruments is characterized by sensors, actuators, wireless connectivity, and on-board processing. These features enable smart musical instruments to directly exchange musically-relevant information with one another as well as communicate with a plethora of external devices (such as smartphones, wearables, virtual reality headsets, or stage equipment). Examples of existing smart musical instruments are the Smart Mandolin [5], the Smart Cajón [6] and MIND Music Labs' Sensus Smart Guitar[1] [7].

According to the vision proposed in [1], a smart instrument is characterized by five core capabilities that define its embedded intelligence: i) knowledge management, i.e., the capability of maintaining knowledge about itself and the environment; ii) reasoning, i.e., the capability of making inferences on the acquired knowledge; iii) learning, i.e., the capability of learning from previous experience; iv) human-smart instrument interaction, i.e., the capability of interacting with the player in ways that extend the bare sound production, such as adaptation and proactivity; v) smart instrument-Musical Things interaction, i.e., the capability of wirelessly exchanging information with a diverse network of interoperable Musical Things.

The sound engine of a smart musical instrument is responsible for the generation of the instrument's digital sounds and may encompass various components (see Fig. 1). For instance, a component can process the sounds detected by a microphone by applying digital audio effects to it; a component can trigger sound samples thanks to a sampler; a component can generate sounds resulting from the control of synthesizers and drum machines; a component can play back different backing tracks. The parameters of each of these components of the sound engine can be modulated by the sensors present in the sensor interface, by means of a set of mapping rules [8]. The sound engine is also responsible for recording the overall sound resulting from the mixing of all such components, but can also record in separate files the contribution of each component.

Smart musical instruments are instances of Musical Things within the "Internet of Musical Things" (IoMusT) paradigm [9], an extension of the Internet of Things [10] to the musical domain. Within this paradigm, smart musical instruments can exchange content with other Musical Things leveraging application and services built on top of the connectivity infrastructure. To date, a topic that has received remarkably little attention in smart musical instruments research is that of defining an interoperable file format for the exchange of content produced by smart musical instruments.

This paper investigates the design of a format specific to smart musical instruments but that at the same time enables interoperability with other Musical Things. To address this topic we adopted a participatory design methodology [11] consisting of a set of interviews with studio producers familiar with the smart instruments concept. The purpose of such interviews was that of identifying a set of use cases for a

[1]https://www.mindmusiclabs.com/sensus/

format encoding data generated by smart musical instruments, with the end goal of gathering requirements for its design.

II. Co-design of use cases

A. Participants and procedure

Two male professional studio producers took part to the interviews. They were aged respectively 40 and 42, and had respectively 17 and 18 years of professional experience as studio producers. Both of them were familiar with the smart instruments concept and had direct experience in using the Sensus Smart Guitar developed by MIND Music Labs. They were also familiar with the Internet of Musical Things paradigm.

One semi-structured interview was conducted for each participant. The interviews were conducted at the respective studios of the producers. Both interviews lasted approximatively one hour and a half. Participants were asked to envision a set of use cases for a new format for smart musical instruments. They were specifically instructed to consider the smart features of smart instruments as well as their needs as studio producers. The interviews were audio recorded and were then analyzed by means of an inductive thematic analysis method [12].

B. Use cases

We conducted an inductive thematic analysis by generating codes from the interviews transcripts. The codes were further organized into themes that reflected patterns for use cases, as described below.

Advanced studio productions. Both participants deemed important that a format encompassing various kinds of information related to a smart instrument's affordances would be useful in contexts of studio production as it could enable novel ways to edit a recording generated by a smart instrument. In addition to the conventional recording of the instrument into an audio file, a new format specific to smart musical instruments may encompass information relating to different aspects of the instrument, such as the sensors signals, different audio tracks generated by the sound engine, or the structure of the sound engine.

A studio producer could interact with these new levels of information in order to create a modified version of the original recording. For instance, the format could enable:

- the mixing of the various audio tracks related to the different components of the sound engine of the smart instrument (e.g., the instrument signal processed with effects, the synthesizers, the samplers, the backing tracks);
- the application of effects to the audio tracks corresponding to the various components of the sound engine;
- the modification of the mappings between sensors values and the parameters of the sound engine (e.g., a sound sample triggered by a sensor in the instrument's sensor interface could be substituted by another sound sample; the sensor associated to a parameter of an effect could be associated to a parameter of another effect).

Smart instrument configuration. Both participants suggested that the format could encompass all information related to the configuration of the smart instrument. By reading the format content, the settings regulating the behavior of a certain smart instrument could be used to configure an instrument of the same kind. For instance, a format saved in a certain smart instrument and loaded into another smart instrument having the same characteristics could configure:

- the structure of the sound engine: this may include which components are present, such as the chain of the sound effects applied to a microphone or the number, type, brand and model of the involved digital synthesizers, samplers, or drum machines;
- the mappings between the sensors and the sound engine parameters;
- a set of parameters regulating the behavior of the instrument, such as the sampling rate of the audio tracks and of the sensor values, the BPM of the backing tracks, the initial values of all the parameters of the sound engine.

Score information. Both participants indicated that the format could encompass the MIDI score of each component of the sound engine. This may include for instance the notes generated by the drum machine, those generated by the synthesizers, as well as the notes generated by the acoustic instrument. The latter could be achieved by means of automatic transcription techniques (see e.g., [13]), and include also information related to a certain playing technique associated to each note (see e.g., [6]). The information about the score could be useful for composition purposes, where the content of the recorded file could be extended with additional tracks composed on the basis of the provided score.

Learning and training. Both participants highlighted that a software for decoding and playing the format (for both the cases in which it is placed inside the instrument and on an external device such as a laptop), could be used for learning and training purposes. For instance, a player of a certain smart musical instrument could load the file format of a particular music piece, mute all (or some of) the interactive tracks (i.e., the track of the recorded instrument and those resulting from the interaction with the sensors) and play over the remaining tracks (e.g., the backing track) to practice the piece in all (or some of) the interactive parts.

Mulsemedia reproductions. Both participants suggested that a software for decoding and playing the format, for instance running on a PC or on a smartphone, could be used in conjunction with Musical Things providing additional sensory content to the music played. This use case relates to contexts of mulsemedia (i.e., multi-sensory media) applications [14] within the Internet of Musical Things paradigm, where a smart musical instrument is used to control in real-time Musical Things aiming at enriching the audience's musical experience with content involving other sensory modalities, such as visual content (e.g., delivered on conventional screens or head-mounted displays) and haptic content (e.g., delivered by musical haptic wearables [15]). The additional content should be perfectly synchronized with the musical content of the file format.
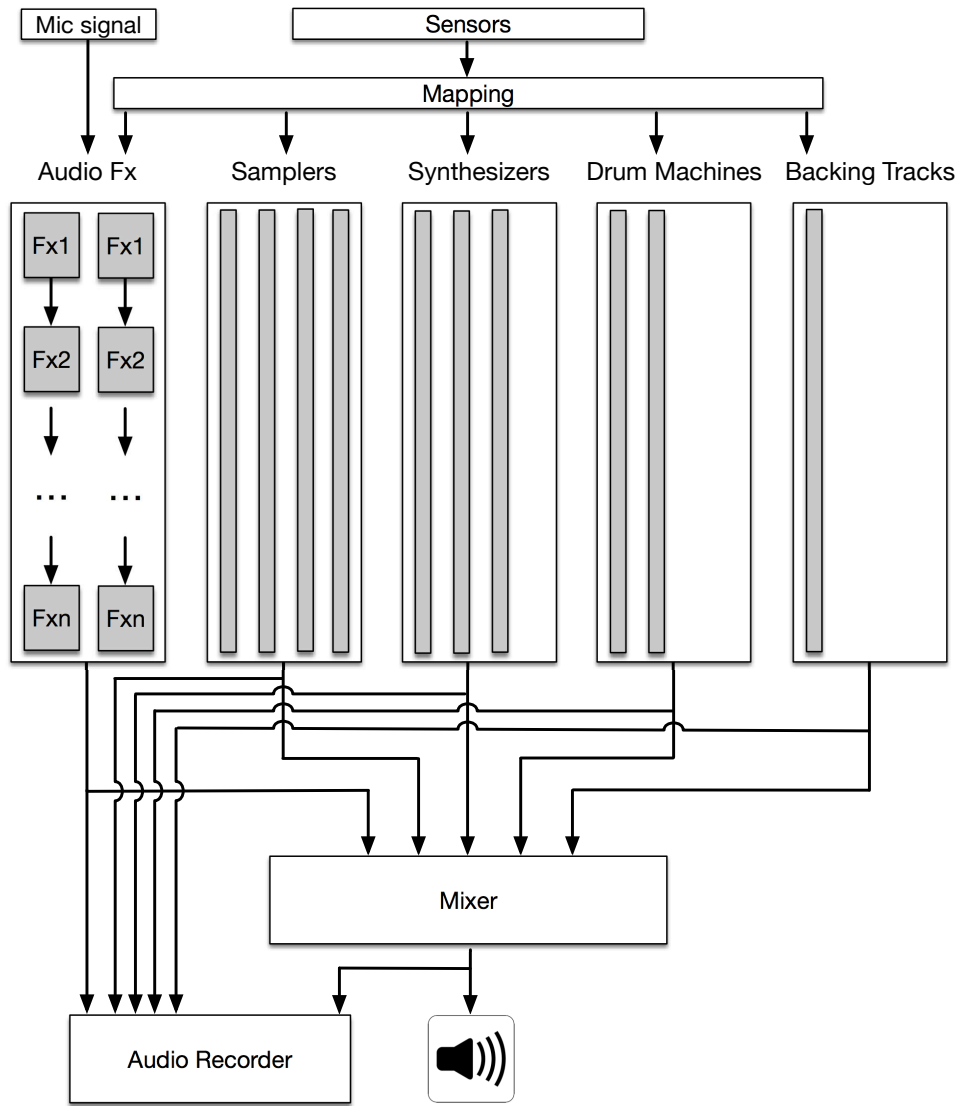
Fig. 1. Block diagram of an example of sound engine running on a smart musical instrument. The gray areas indicate the various sub-components (e.g., separate tracks of effects chains or different parallel instances of synthesizers).

III. REQUIREMENTS

The five use cases resulting from the interviews were analyzed with the purpose of identifying a set of requirements that the file format should have. Such analysis resulted in the following design requirements for the specification of the format.

The format shall contain:

- **audio tracks:** these are related to the outputs generated by each of the components of the sound engine as well as the overall output.
- **sensors tracks:** the values of each sensor (both analog and digital) present in the sensor interface are represented;
- **static metadata:** these are related to the instrument and its configuration, including i) type, brand, and model of the instrument; ii) information about the sensor interface (including how many sensors, which type); iii) the structure of the sound engine (including which components are present, the chain of effects, the brand and model of the audio plugins utilized); iv) the sampling rate for audio and sensors tracks;
- **dynamic metadata (i.e., time-based):** these are related to i) the sensors-to-sound parameters mapping, including the range of variation of each controlled parameter, the type of mapping (e.g., one-to-one, one-to-many, etc.), and the type of mapping function (e.g., linear, logarithmic, exponential, ad-hoc, etc.); ii) which subcomponents of each components are active at a given time (e.g., in a section of a musical piece a synthesizer is active, while in another section it is not); iii) the MIDI score of the piece, including additional information related to each note such as the type of gesture that generated it; iv) the BPM of the backing track.

The format shall support:

- **mixing:** the audio tracks can be muted, removed, substituted, as well as their individual volume can be adjusted and effects can be applied to them;
- **change of structure of the sound engine:** each of the components and subcomponents of the sound engine can be modified (e.g., a certain sound effect plugin can be substituted with another one);
- **change of mapping:** each of the mappings can be modified in all its parts, including the substitution of the controlled parameter and its rage of variation as well as the associated mapping function;
- **interaction with other Musical Things:** the reproduced audio content should be synchronized with the multisensory content delivered by connected Musical Things.

IV. DISCUSSION AND CONCLUSIONS

Despite their crucial role for enabling interoperability, today standardization activities are in vast part unrealized within the Internet of Musical Things paradigm [9]. The requirements reported in this paper represent a preliminary step towards the definition of a file format for the emerging family of smart musical instruments. A format for the exchange of content generated by a smart instrument would enable novel interactive applications such as those of the use cases co-designed with the two studio producers involved in the interviews. The existing standardized formats that are closest to meet the requirements presented in section III are the IEEE 1599 [16], [17] and the IM AF (MPEG-A: Interactive Music Application Format) [18].

IEEE 1599 is a format encoding audio recordings and synchronized XML symbols describing the musical information. Specifically, it is composed of six layers: i) general: it contains metadata relevant to the entire document; ii) logic: it describes the score symbols; iii) structural: it describes the musical objects and their interactions; iv) notational: it contains the graphical representation of the score; v) performance: it contains computer-based descriptions of a musical performance; vi) audio: it contains the digital audio recording.

IM AF is a multitrack format that allows users to mix individual tracks for different musical instruments by adjusting their volume, as well as enables the association of synchronized text (e.g., for lyrics). Moreover, it supports a set of presets for the mixes created by the producer as well as a user mixing mode (optionally encompassing interactivity rules). Specifically, an IM AF file consists of: i) multiple audio tracks: they contain the audio signal related to each instrument or voice, and may be encoded either in 2D or 3D spatial audio; ii) groups of audio tracks: the definition of the structure of the audio tracks into groups; iii) preset data: a set of predefined mixes of the multiple tracks; iv) user mixing data and interactivity rules: this information relates to the interaction of the user with the mixing parameters in the file; v) metadata: this is data related to the music contained in the file and can be both static (e.g., information about the song or the album,

and still pictures) and time-dependent (e.g., the synchronized lyrics).

The structure of an IM AF file is derived from the MPEG-4 ISO-Base Media File Format standard (ISOBMFF)[2], the most widely deployed standard in the music[3] [19] and media industry (aka .mp4). Another standard that is ISOBMFF-compatible is MPEG-V[4]. This format is conceived to support, among other things, the encoding of sensors data.

The pro and cons of the IEEE 1599 and IM AF are discussed in [20]. To fulfill the identified requirements both formats require an extension that includes novel features. Compared to IEEE 1599, IM AF supports multitracks and has the advantage of supporting the integration with features from ISOBMFF, which ensures interoperability with ISOBMFF-based applications. These features may be enhanced timed-metadata such those of MPEG-V encoding the sensors information. For example, while IEEE 1599 has native support for MIDI, IM AF getting advantage from ISOBMFF timed-metadata features, in the same way in which it supports synchronized lyrics, it could be used to encode not only the temporal evolution of sensors and mappings but also MIDI information.

Differently from IEEE 1599, IM AF supports sets of rules satisfying the artist's requirements related to the users' possibilities for mixing the various tracks (i.e., removing, muting, or adjusting the volume of a given track or group of tracks). However, such rules are not designed to support the logic underlying the mappings from sensors values to sound parameters, therefore a new set of rules would be required to handle such mechanisms.

Starting from the requirements identified in this paper, in future work we plan to design and implement an encoder and a decoder for a format for smart musical instruments. Following the comparison between features, capabilities, and market deployment of IEEE 1599 and IM AF, we plan to implement the identified requirements by integrating into IM AF the aforementioned features from ISOBMFF. Notably, our implementation will need to encode the configuration of the smart instrument. For this purpose, an interesting avenue that can be explored is that of leveraging ontologies able to represent the knowledge about the sound engine, such as the audio effect ontology [21] and the studio ontology [22]. Furthermore, it is worth noticing that this paper focused on a format for one single instrument while the focus could be also directed towards the exchange of files containing content from multiple smart musical instruments. All these matters will also be object of our future investigations.

---

[2]ISO/IEC 14496-12:2015 "Information technology – Coding of audio-visual objects – Part 12: ISO base media file format (ISOBMFF)"

[3]See e.g., LANDR by MixGenius (2017): https://en.wikipedia.org/wiki/LANDR and https://youtu.be/K5CsEUIh_GI

[4]ISO/IEC 23005, "Information technology – Media context and control (MPEG-V)"

REFERENCES

[1] L. Turchet, "Smart Musical Instruments: vision, design principles, and future directions," *IEEE Access*, 2018.

[2] E. Miranda and M. Wanderley, *New digital musical instruments: control and interaction beyond the keyboard.* AR Editions, Inc, 2006, vol. 21.

[3] E. Berdahl, "How to make embedded acoustic instruments," in *Proceedings of the Conference on New Interfaces for Musical Expression*, 2014, pp. 140–143.

[4] C. Rottondi, C. Chafe, C. Allocchio, and A. Sarti, "An overview on networked music performance technologies," *IEEE Access*, vol. 4, pp. 8823–8843, 2016. [Online]. Available: https://doi.org/10.1109/ACCESS.2016.2628440

[5] L. Turchet, "Smart Mandolin: autobiographical design, implementation, use cases, and lessons learned," in *Proceedings of Audio Mostly Conference*, 2018.

[6] L. Turchet, A. McPherson, and M. Barthet, "Real-time hit classification in a Smart Cajón," *Frontiers in ICT*, vol. 5, no. 16, 2018. [Online]. Available: https://doi.org/10.3389/fict.2018.00016

[7] L. Turchet, M. Benincaso, and C. Fischione, "Examples of use cases with smart instruments," in *Proceedings of Audio Mostly Conference*, 2017, pp. 47:1–47:5. [Online]. Available: https://doi.org/10.1145/3123514.3123553

[8] A. Hunt, M. Wanderley, and M. Paradis, "The Importance of Parameter Mapping in Electronic Instrument Design," in *Proceedings of the Conference on New Interfaces for Musical Expression*, 2002.

[9] L. Turchet, C. Fischione, G. Essl, D. Keller, and M. Barthet, "Internet of Musical Things: Vision and Challenges," *IEEE Access*, vol. 6, pp. 61 994–62 017, 2018.

[10] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010. [Online]. Available: https://doi.org/10.1016/j.comnet.2010.05.010

[11] D. Schuler and A. Namioka, *Participatory design: Principles and practices.* CRC Press, 1993.

[12] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, 2006.

[13] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: challenges and future directions," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

[14] G. Ghinea, C. Timmerer, W. Lin, and S. Gulliver, "Mulsemedia: State of the art, perspectives, and challenges," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 11, no. 1s, p. 17, 2014.

[15] L. Turchet and M. Barthet, "Co-design of Musical Haptic Wearables for electronic music performer's communication," *IEEE Transactions on Human-Machine Systems*, 2018 (In press).

[16] L. Ludovico, "Key concepts of the IEEE 1599 Standard," in *Proceedings of the IEEE CS Conference The Use of Symbols To Represent Music And Multimedia Objects*, 2008, pp. 15–26.

[17] A. Baratè, G. Haus, and L. Ludovico, "A critical review of the IEEE 1599 standard," *Computer Standards & Interfaces*, vol. 46, pp. 46–51, 2016.

[18] I. Jang, P. Kudumakis, M. Sandler, and K. Kang, "The MPEG Interactive Music Application Format Standard [Standards in a Nutshell]," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 150–154, 2011.

[19] M. Le Goff, C. Carrier, and S. Walker, "Introducing stem: a new multichannel audio format," in *Proceedings of International Society for Music Information Retrieval Conference*, 2015.

[20] R. Stewart, P. Kudumakis, and M. Sandler, "Interactive music applications and standards," in *International Symposium on Computer Music Modeling and Retrieval*. Springer, 2010, pp. 20–30.

[21] T. Wilmering, G. Fazekas, and M. Sandler, "The audio effects ontology." in *Proceedings of the International Society for Music Information Retrieval conference*, 2013, pp. 215–220.

[22] G. Fazekas and M. Sandler, "The Studio Ontology Framework," in *Proceedings of the International Society for Music Information Retrieval conference*, 2011, pp. 24–28.