# A Proposal for Standardization of Internet of Musical Things (IoMusT) Environments


Rômulo Vieira
*MídiaCom Lab*
*Fluminense Federal University*
Niterói, Brazil
romulo_vieira@midiacom.uff.br

Flávio Schiavoni
*Arts Lab in Interfaces, Computers, and Everything Else*
*Federal University of São João del-Rei*
São João del-Rei, Brazil
fls@ufsj.edu.br

Débora C. Muchaluat-Saade
*MídiaCom Lab*
*Fluminense Federal University*
Niterói, Brazil
debora@midiacom.uff.br



***Abstract*—The Internet of Musical Things is an interdisciplinary area of knowledge that aims to improve the relationship between musicians, as well as between musicians and audience members, favoring concerts, studio productions, and music learning. Although emerging, this field already faces some challenges, such as lack of privacy and security, and mainly, lack of standardization and interoperability between its devices. Therefore, the present paper presents an environment design, called Sunflower, highlighting its architecture, protocols, and sound features that can contribute to solving the most recurrent problems in this area.**



***Index Terms*—Internet of Musical Things; network music performance; participatory art**


## I. Introduction

The expansion and cheapening of broadband Internet, data storage media, and electronic devices that have been observed since the beginning of the 21st century culminated in the emergence of the Internet of Things (IoT), whose basic concept deals with the widespread presence of a variety of objects in everyday life, which through unique addresses, can interact with each other and cooperate with their neighbors to achieve common goals [1].

When IoT domains are expanded to musical practice, a field called the Internet of Musical Things emerges. This area is characterized by being multidisciplinary and can be described as a set of interfaces, protocols, and representations of information related to music, which enable services and applications with an artistic purpose from interactions between humans and musical things or between musical things themselves [2].

A musical thing, in turn, can be defined as an electronic device capable of acquiring, processing, performing, or exchanging data that serve a musical purpose [2]. The combination of these devices with services and sound applications creates an interoperable environment, responsible for interconnecting musicians, instruments, and audience members, which multiplies the possibilities of interaction in art shows and begins a relationship of interdependence between the elements [3].


The authors would like to thank the support of the funding agencies CNPq, (Grant Number 151975/2019-1), CAPES (Grant Number 88887.668167/2022-00), FAPEMIG, UFSJ, and FAPERJ.


Despite being an area with great artistic and computational potential, IoMusT lacks standardization in its environments, largely because it is a novelty. As a result, some work has emerged in the area to tackle this problem. However, they are fragmented, focusing mostly on how communication takes place over the network and how the structures of musical things are, leaving aside a discussion about the high-level control of these devices. In addition, most environments use concepts and technologies that follow the preferences of their developers, not allowing them to be adapted to the most varied usage scenarios. All these difficulties and issues motivated the elaboration of an environment design, called Sunflower, which presents an operating mode based on the Pipes-and-Filters architecture and which aims to mitigate the problems of lack of interoperability in this area.

The remainder of this paper is structured as follows. Section II presents the Sunflower environment, addressing its architectural structure and layering division. In addition, it addresses its protocol for exchanging messages. Next, Section III given its implementation and the results obtained from the practical tests carried out. Finally, discussions about the results obtained and final considerations are presented in Section IV.

## II. The Sunflower environment

The design of a system consists of presenting its architecture, modules, interfaces, protocols, data, design process and decision making that help in the execution of a given task [4]. The model proposed for Sunflower aims at a new way of thinking about IoMusT environments and indicating possible solutions to the most recurrent problems in this area.

As for the desirable features for the ecosystem, it is difficult to reach a consensus for them, since there are few examples to support these decisions. In this way, the authors used the definitions offered by the areas that permeate the IoMusT [2].

From the desirable features for the devices, the following stand out: heterogeneity and transparency, graphical display for exchanging control information, input and output information, and integration with legacy software and hardware [3].


2022 IEEE 8th World Forum on Internet of Things (WF-IoT) | 978-1-6654-9153-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/WF-IOT54382.2022.10152123

### A. Thinking about Layers

Several tools in the area of computer science use layered architecture, such as operating systems and the TCP/IP model. Generally, this framework is used to isolate implementation details from one layer to another, providing some maintainability for the system. In such a way, the levels in Sunflower are also independent, being characterized according to the musical things, data types, and protocols present in each of one them. The layers are: **digital audio layer**, responsible for generating sound data for the environment; **graphic layer**, which allows the system to deal with visual representations; **control layer**, which provides to the environment the ability to control volume, frequency, and beats per minute (BPM) in a remote way; and finally, **management layer** that allows the system administrator to visualize who is connected to it and what its main features are, admitting that it only interconnects those that can exchange data with each other.

### B. Protocol Proposal

In Sunflower, each entity at each layer uses a **setup protocol** to implement its service definitions, divided into 5 steps: i) discovery, search for nodes on the network and establish contact between them; ii) device setup, allows eventual changes in the network settings and/or device data to occur; iii) setting up and connecting the environment, exchanges messages that inform its credentials and ports used in the communication. If the connection is confirmed, the system shows its new status. Otherwise, it indicates that the attempt was denied; iv) change mode, allows switching to “performance mode”, where devices no longer accept remote settings; and v) close the connection, definitively disconnects the musical things.

## III. Implementation

Considering the structure and protocols that govern the operation of Sunflower, its four layers were conceived, in a practical way. The system has 27 prototypes of musical things. They were developed in Pure Data, chosen for being multiplatform, open-acess, easy to program, and for being present in several systems that deal with music over the network and/or in real-time. To further expand the artistic and interactive possibilities of this system, four arts were created in Processing.

In the control layer, every musical thing can be managed by the network. To fulfill and manage these and other requirements, a Command Line Interface (CLI) was created in Python. From a functional point of view, the CLI presents information about the IP address and ID number of each musical thing, as well as its input and output data.

Finally, three musical things (drum machine, microphone, and tuning fork) were transposed to smartphones to be played in the MobMuPlat mobile app. In this way, users gain a new platform to interact with the environment, while Sunflower accepts an easy and general-use device, allowing the integration of more people.

The tests were performed on localhost, Ethernet cable and Wi-Fi, using an Acer Predator Helios 300 laptop (computer running Linux Ubuntu Studio 20.04) and a Dell Inspiron 14 3442 3000 series (computer running Linux Ubunbtu Studio 20.04 LTS). A Behringer C1 condenser microphone, a 6-string electro-acoustic guitar, and a Behringer U-Phoria UMC202HD audio interface were also used.

## IV. Discussion and Final Remarks

On localhost, the latency and jitter initially observed were the same as on the system itself, regardless of the network delay. This can be thought of as the time it takes to pack the audio data, copy it to kernel space, pass it back to user space, and finally unpack the data. On the other hand, video data, as it consumes more computer and network resources, travel more slowly between devices. The OSC control data, being small in the number of bytes and sent spaced out, was received virtually instantaneously.

On the Ethernet connection, an average latency of 2.083 ms could be observed, while the jitter was 0.761 ms. On the Wi-Fi connection, the latency was 4.453 ms, and the jitter was 1.843 ms, confirming that the wired connection is faster than the wireless one. As the data exchange rate was constant, the first 50 packets were considered for estimation purposes.

An important point to be highlighted in the tests, especially wireless ones, is that as the number of devices in the network increases, so does its latency. Also in this sense, it is noteworthy that the tests present user limits according to the physical and link layers. The implementation in a minimized scenario served to guide this communication and show the Sunflower concepts applied in practice, but if it happened in a more realistic scenario, the network infrastructure would need to be improved.

Sunflower comes up to solve interoperability and heterogeneity issues in IoMusT environments, allowing devices to establish communication with their neighbors without the need for prior knowledge of their IP number, in addition to avoiding communication restrictions due to the physical features of each one. There is also a layered division that guarantees the system reusability and ease of expansion and maintenance.

Furthermore, it proposes a way of functioning that is independent of tools, with an unprecedented discussion for this field of research. Even so, it is not intended to end the discussion about which audio file formats and protocols should be used in IoMusT communication, in the same way that the authors do not claim for themselves the authority to say what should or should not be done when planning an environment in these molds.

## References


[1] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer Networks*, pp. 2787–2805, 10 2010.

[2] 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, 09 2018.

[3] R. Vieira, L. Gonçalves, and F. Schiavoni, “The things of the internet of musical things: defining the difficulties to standardize the behavior of these devices,” in *2020 X Brazilian Symposium on Computing Systems Engineering (SBESC)*, 2020, pp. 1–7.

[4] B. Burns, *Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services*, ser. 1. O’Reilly, 2018, no. 1.