Japan Bilingual Publishing Co.
Cultural Arts Research and Development
http://ojs.bilpub.com/index.php/card

ARTICLE

Internet of Musical Things Environments and Pure Data: A Perfect Match?

Rômulo Augusto Vieira Costa*   Flávio Luiz Schiavoni
Federal University of São João del-Rei, Brazil

ARTICLE INFO

Article history
Received: 28 December 2021
Accepted: 08 February 2022
Published Online: 18 March 2022

Keywords:
Pure data
Internet of Musical Things
Sunflower environment

ABSTRACT

The Internet of Musical Things (IoMusT) is an emerging area based on concepts from the Internet of Things, new interfaces for musical expression, human-computer interaction, ubiquitous music, and many others. Despite being a novelty, it already has some ecosystem models that use their abstractions to allow musical practice. However, they are not standardized and tend to consider only the preferences of their creators, relegating Pure Data, when present, to a secondary role. In this context, this article presents the Sunflower environment, a tool created by the authors of this text and which uses Pure Data in all its development stages, highlighting the importance of such a tool.

1. Introduction

Music has always been susceptible to the most diverse technological changes that accompanied each era, and this became clearer in the progress that occurred in the form of recording and reproduction of musical content. In such a way, these advances can be divided into four distinct seasons [1-3]. The first of them, called the "Acoustic Era" (1877 – 1925), was characterized by the exclusive use of mechanical devices for recording.

Then came the "Electrical Age" (1925 - 1945), which from the use of electricity, as the name indicates, allowed the creation of new artifacts that helped this process, such as microphones, amplifiers, and recorders, which favored the emergence of popular music genres such as jazz, blues, and rock. In the 1940s, the emergence of the vinyl record marked the beginning of the music industry, besides creating a market logic, where phonograms were marketed in the form of singles and albums, with title, cover, and booklet, a standard that continues today.

After World War II, magnetic tapes began to be used for recording audio, beginning the "Magnetic Era" (1945 – 1975). The creation of the cassette tape, in the 1960s, was perhaps the most striking fact of this period.

The fourth and final season is the "Digital Era", which began in 1975 and continues to this day. By replacing analog sound with digital encoding, it has quickly and dramatically changed the way music is created and

*Corresponding Author:
Rômulo Augusto Vieira Costa,
Federal University of São João del-Rei, Brazil;
Email: romulo_vieira96@yahoo.com.br

DOI: https://doi.org/10.55121/card.v2i1.13

consumed. The emergence of the Compact Disc (CD) marked this change. However, the biggest transformation took place in the 1990s, with the spread of the internet and acoustic compression techniques, which culminated in the MP3 format, which eliminated the need for physical media to hold music files.

From the turn of the century, various technological changes continued to impact the world. Among them, we can mention the popularization of broadband internet and the cheapening of data storage media and electronic producer, which culminated in the emergence and expansion of an area that was called the Internet of Things (IoT).

This concept can be defined as the widespread presence of a variety of objects that are able to interact with each other and cooperate with their neighbors, through unique network addresses, to achieve common goals [4,5]. When its concepts are expanded to musical practice, a new area emerges called the Internet of Musical Things (IoMusT).

The Internet of Musical Things (IMusT) is a multidisciplinary research field, composed of concepts from the Internet of Things, ubiquitous music, new interfaces for musical expression, human-computer interaction, artificial intelligence, and many others. From a technical point of view, it refers to a network of devices dedicated to the production or reception of musical content, called musical things, which are defined as:

Entities that can be used in a musical context to produce or observe phenomena associated with this type of practice, which can be connected to a local and/or remote network and acting as a sender and/or receiver. A musical thing can be, for example, an intelligent instrument (a new family of instruments that uses sensors, actuators and wireless connection for sound processing) [6], a wearable tool (devices that the user wears or uses as an accessory, which can capture information from the environment and help immersion in a live performance) or any other networked device used to control, generate or track responses to musical content [7].

It is also noteworthy that musical things are digital elements, which can be programmed and connected to the internet, they must behave like conventional devices, created to perform a set of well-defined tasks and not meet a general-purpose, and how they are used in artistic activities, aesthetic and ergonomic factors are as valuable as technical [8].

These devices can be categorized in the same way as the elements traditionally used in musical practice, once the peculiarities of this category of equipment are observed. In this way, they are classified into: audio equipment, with the bonus of allowing the sending and receiving of data over the network, and also remote control through smartphones; musical instruments, which function similar to traditional instruments, but are equipped with sensors and actuators that react to environmental stimuli, helping with performance; and tools that help music practice, transporting equipment such as tuners and metronomes to the digital medium, or even controlling the lights via the network [8].

Despite being standalone devices, they are not useful on their own, requiring inclusion in an ecosystem. These ecosystems include other interoperable devices and services that connect musicians, artists, sound engineers, teachers, students, and audience members to support their interactions as well as expand their musical and artistic capabilities. Figure 1 summarizes this behavior [9,10].

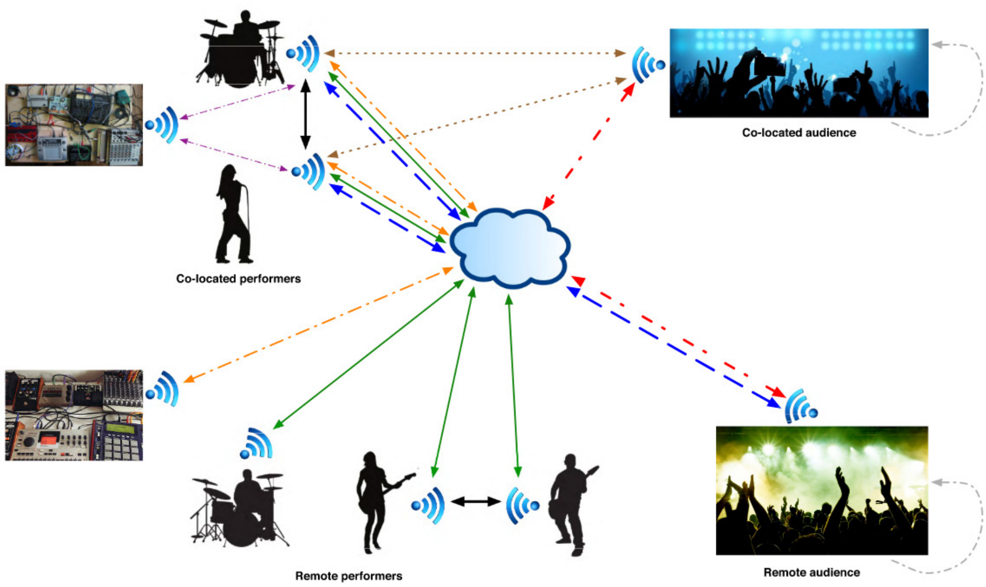

Figure 1. Interactions in an IoMusT environment [9].

In order to have a good functioning of the environments that are generated from the interconnection of multiple musical things and users, they must have low latency, synchronization, interoperability, scalability, and easy integration by the users. Added to this, it must be able to handle different types of signals, such as audio, video, and control, as well as assisting in the creative process. It is in this context that the main contribution of this article arises, by proposing Pure Data as a central tool in the development of these systems, given that the language can handle all types of signals mentioned, in addition to the ability to send data natively over the network, be cross-platform and open source. This approach is further discussed in Section 1.

As with any network connection, this environment also requires a standardization of the data that will navigate through it. Although the IoMusT area is promising and proposes changes in music-making, it fails to deal with the heterogeneity and standardization in its systems. Although there are efforts to solve these problems, the environments developed so far considered only the technologies and preferences of their creators, relegating to Pure Data, when present, only the role of an audio engine. In this sense, the authors of this work present in Section 2, the Sunflower, an environment developed by themselves and which counts on Pd as a central tool to obtain the characteristics desired by IoMusT, while possible use scenarios are presented in Section 3.

Section 4 is responsible for presenting a discussion about the use of Pd in IoMusT ecosystems, in particular the Sunflower, debating how it inspired its modeling and made its construction feasible. Finally, Section 5 displays a brief summary of the topic analyzed in this text, as well as explains its relevance and demonstrates summarized conclusions about the performance of this work.

2. Intersections between IoMusT environments and Pure Data

When analyzing the proposals for existing IoMusT environments, we found Pure Data acting as the system's audio engine, being the tool responsible for creating the sound and/or capturing the audio data and sending them over the network. This is due to the fact that the language is cross-platform and extensible, being able to run on smartphones and microprocessor boards, such as Raspberry and Bela, and because it allows users to insert features created by themselves, which guarantees certain extensibility to the system.

However, we believe that its use goes beyond this simple task, being the tool that links artistic making to the essential characteristics predicted by the Internet of Musical Things. Although some connection aspects are more related to the network than to the means to access it, the native support that Pd offers reduces system latency and supports data synchronization, as it is not necessary to use a tool third to send data over the network. These two factors, added to the reprogramming capability, code reuse, and real-time language update, allow for the simple and quick creation of new patches, which guarantees the scalability of the system. This also reflects on the integration by the users, who do not need to enter the network with a new code and can benefit from one created previously. In addition, with a few commands, a user without prior knowledge of the language can already generate some action that will impact the environment.

Another important point of Pd concerns the fact that it supports several audio and video files, as well as music information control protocols, such as MIDI and OSC. Therefore, it ensures another important characteristic of IoMusT, which is the variety in the data flow of its systems, in addition to allowing the integration of different devices that can handle this range of information, or even those that are legacy, increasing the heterogeneity of the system.

Together, all these features will reflect on the creative process, where musicians and users with different abilities can participate together in the same environment.

3. Materials and Methods

Given all the advantages that Pd can offer beyond a simple sound generation tool, the authors developed an environment centered on this language, called Sunflower, with an operating mode based on the Pipes-and-Filters software architecture and a division in layers, which in addition to contemplating important characteristics that this ecosystem must present, also act to reduce the problems arising from the heterogeneity and lack of standardization of the data that travel through it. More details on the implementation and functioning of Sunflower are provided below.

3.1 A Model to Think an IoMusT Environment: The Pipes-and-Filters Architecture

IoMusT environments are formed by autonomous musical things, with different structural characteristics and data, which have never had prior contact. It is also noteworthy that when planning this archetype of the network, the number of users who can connect to it should not be limited. Thus, it is necessary to think of a system architecture that is capable of connecting these different objects. Therefore, the authors propose a way of functioning analogously to the Pipes-and-Filters architecture.

This model is composed of filters, responsible for receiving a stream of data in their inputs, processing them according to the logic of one or more algorithms, and sending the results for output using communication channels, called pipes. It is worth noting that filters are independent components, that is, they do not relate directly to their peers, restricting their functionality only to what is received at the input and what will be observed at the output, allowing them to be allocated in different positions, in order to get different results in the outputs [11,12].

In the environment proposed here, musical things behave in an analogous way to filters, without prior knowledge of their neighbors and using as a communication reference only what they can receive as inputs and outputs. They are further classified according to their mode of operation, being source, when they supply only data to the system, sink, when they only consume this data, or mixed filter, when they perform both functions. They can use different network transport protocols, such as TCP or UDP, as well as broadcast or multicast addressing.

The pipes, on the other hand, are cables common to musical practice, such as the External Line Return (XLR), or the means of transmission over the network, such as twisted-pair cable and Wi-Fi. In some situations, it may be necessary for them to store buffers, but never apply processing or modifications to the data that navigate through them.

It is important to highlight that the Pipes-and-Filters model only inspired the behavior of Sunflower, as this is an approach used for software development and Sunflower is a network communication structure. That said, data exchange takes place through the client-server paradigm, where sink filters play a role similar to that of clients, as they only consume data, and source filters are the servers, responsible for providing the requested information. The advantage of thinking about the system in this way is that a single musical thing can be connected to several other devices, proposing a one-to-many mapping, a counterpoint to the traditional model of connecting objects in musical environments, which offers a wide domain of mapping one-to-one.

However, the use of such strategy gives rise to three problems: i) diversity in the data format; ii) occurrence of possible overloads; and iii) data incompatibility can make filter reuse more difficult.

To solve these problems, Sunflower was divided into layers, a very common approach in Computer Science, used to isolate the implementation details of each layer and facilitate its reuse or possible maintenance. However, the interpellation proposed here is a little different, as the layers are not necessarily interdependent, where their parallel functioning allows the influence of one on the other. This metaphor was adopted to indicate that there is a separation of musical things, data, and protocols present in each one of them. That said, the ecosystem has been divided into four better layers, detailed below.

3.2 Digital Audio Layer

The audio layer, as the name implies, is responsible for generating sound data, supporting Pulse-code modulation (PCM) audio and WAV files. To ensure the interoperability of the data circulating in this layer, it must provide the system with information about the representation of the audio, informing the sampling rate, number of channels used for communication, bit depth, the format of the file being sent, ports that you use for communication and others.

This layer also supports a multitude of musical things, whether developed by a single user, whether they function similarly to traditional instruments or audio devices, as categorized in Section 1. Thus, effects pedals, recorders, microphones may be present, guitars, drum machines, and many others. Pure Data is a tool that fits this issue, as it allows both the creation of a new object and the simulation or insertion of instruments in the environment. Figure 2 displays the patch responsible for capturing audio from an external microphone and sending its data over the network.

Another important point in this layer is the possibility of dividing the processing between several devices, adding or removing them from the network to guarantee the system's scalability and the creation of music in a shared way. This can reflect on the interactions that occur in artistic presentations, installations, studios, in the band's setup, orchestras, and the like.

3.3 Graphic Layer

The graphic elements provide numerous improvements for a musical presentation, from the video, used to capture images in real-time, highlight the musicians' actions that cannot be observed with the naked eye or even to create visual effects that make up the show, as animations and figures corroborate the aesthetics of the show, while visual information can also be conveyed to enhance interaction and communication between musicians and audience members. For this to occur, the image must go through an encoding and decoding process, so that all visual data are

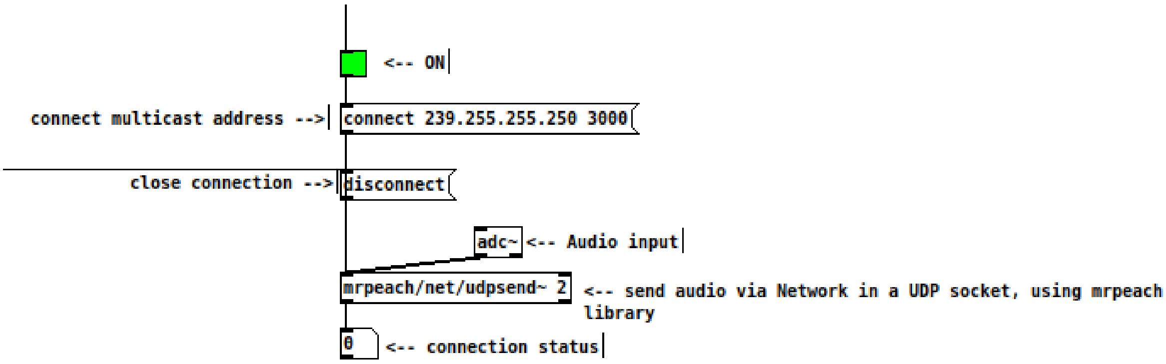

Figure 2. Patch present in the digital audio layer, responsible for capturing audio from a microphone.

Source: The Author.

converted into bits and can be understood by computer systems.

It is in this context that the graphic layer of Sunflower appears, responsible for carrying out this procedure and guaranteeing the environment the ability to work with this category of data. For this, the authors used Pure Data's multimedia capability to accomplish this task, using one of its externals, the Graphics Environment for Multimedia (GEM), capable of reproducing data from DVD players, webcams, laptops or any other means with compatible input and responsible for real-time image processing and synthesis.

This layer also increases audience participation in the show, as in addition to being another controllable functionality, receiving insertions and graphical modifications, changes in tonality, resolution, and the like, it can also provide them with information such as scores and suggestion notes, helping in musical participation. The patch responsible for capturing webcam video and sending it over the network is depicted in Figure 3. As with the audio layer, the UDP protocol is also used.

3.4 Control Layer

Synchronization between devices and the exchange of data between them is at the heart of network communication, essential concepts also for musical practice, where musicians synchronize their actions based on time. Thus, each element present in the environment, be it audio or video must be able to send/receive data and control over the network. This means that they can have properties such as volume, frequency, and Beats Per Minute (BPM) changed remotely, as well as being turned on or off in the same way.

Pure Data plays a crucial role in this layer thanks to its ability to build graphical interfaces that use bangs, toggles, and sliders to control objects and send this information natively over the network. As a distributed system, patch processing can start with one musical thing and end with another.

Such procedure could be performed through text files,

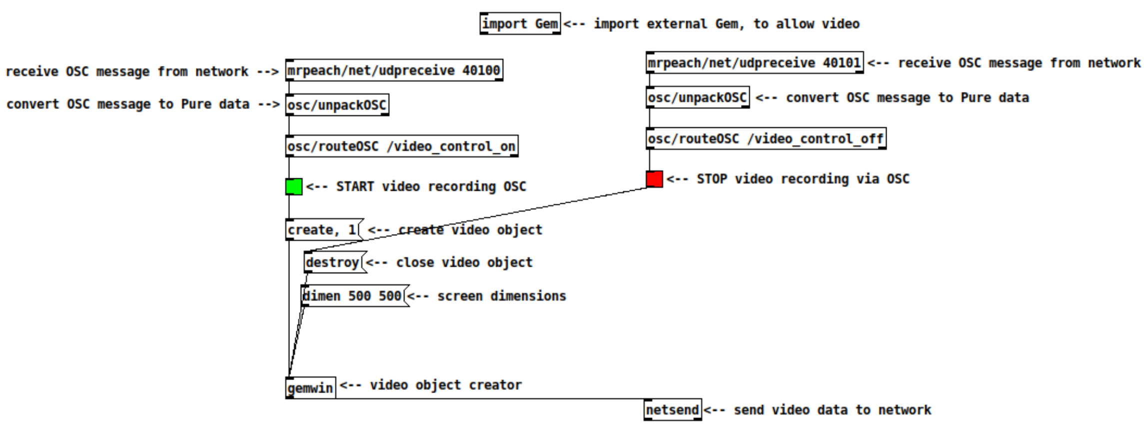

Figure 3. Patch responsible for video capture.

Source: The Author.

Extensible Markup Language (XML), or some other, but the chosen one to perform this function was the OSC music information control protocol. The choice of this protocol over the native types of Pd, such as number, array, and text, was made to ensure the interoperability of the system, allowing communication with other tools and platforms. Also, it was useful to encapsulate MIDI messages and send them over the network, allowing legacy instruments and software, that is, those that left the production line or are no longer up-to-date, to be inserted into the environment, increasing the range of musical things and possibilities for users.

Figure 4 shows the drum control patch, made in Pure Data and which sends OSC control to separately handle the volume of each component of this instrument, as well as change its default, its BPM, turn it on or turn it off remotely.

3.5 Management Layer

In musical performances, the presence of the technician or sound engineer is essential to provide support to the musicians. Similarly, computer networks require an administrator, responsible both for assisting users and for configuring and maintaining the infrastructure. In the Sunflower ecosystem, these two roles are inseparable, with the administrator being responsible for ensuring the usability of the network and also taking care of graphic and sound aspects.

For this, it is necessary that the devices present in the environment can be mapped, where their main characteristics are within reach of the administrator. However, this task can be complex, as many of the devices present may not have a graphical interface, processing unit, or some input and output device that provides this information. Above all, shows open to the public generate unpredictability regarding the number of participants, making it even more difficult to map objects.

Added to this, there are problems arising from the heterogeneity of the environment, which aggregates devices with different formats in audio files, sampling rates, bit depth, video resolutions, color systems, communication protocols, control, etc.

One way to control this is through a configuration through the network, where the administrator can see who is connected to the environment and what its characteristics are, and from there, make the manual connection of those objects capable of exchanging information. For this, every musical thing, when connecting to the network, must publish its resources, such as ID number, ports it uses to send or receive data, audio and video resolutions, protocols, and any other information that is relevant. When disconnecting, they must also post a message with this information.

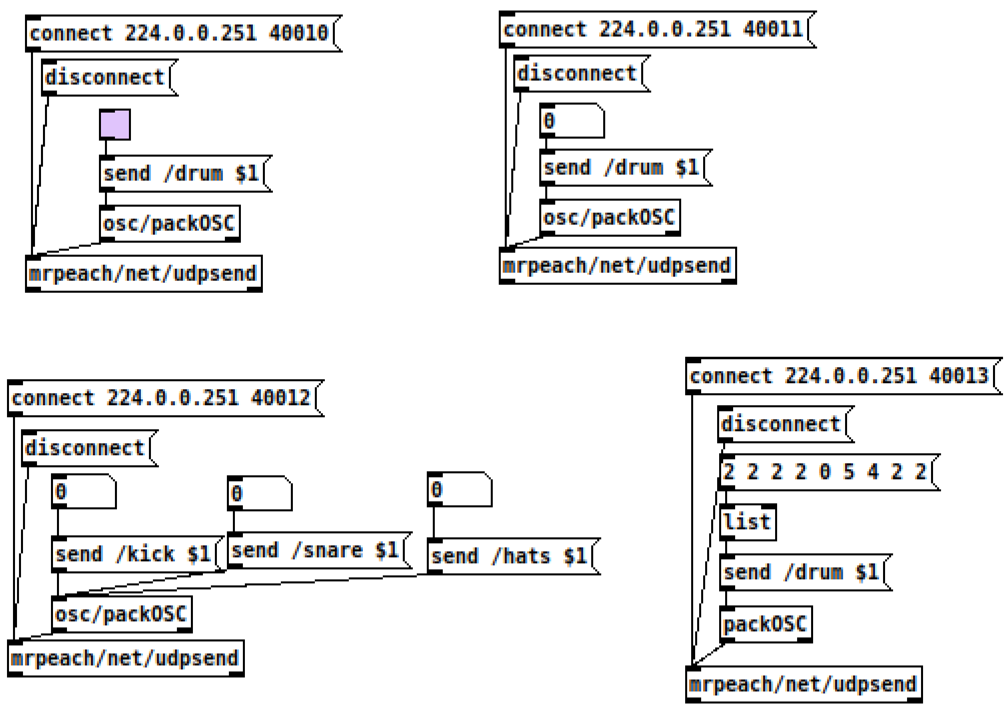

Figure 4. Drum machine patch control.

Source: The Author.

Figure 5, therefore, shows the excerpt of the microphone patch responsible for sending its main information to a Python script, which agglutinates this data in one place and makes the administrator's job easier. As it is not the focus of this work, more information about this script will not be provided.

As stated, data from one layer can interfere with the other, just as a musical thing can meet the requirements of one or more layers. Likewise, a device can act as source in one context and sink in another, in addition to those that are naturally hybrid. Among the patches created are an audio player, drum machine, a patch that receives a connection from a guitar or bass, speaker, patch that receives a connection from an external microphone, pitchfork, recorder, video sender, video receiver, and volume. Also, there is a control patch via the network for each of the mentioned ones, respecting their peculiarities and functionalities.

Even though it is in the development stage, the management layer appears as a key factor in the environment, as it is responsible for displaying the characteristics of each element connected to it and indicating which ones are capable of being connected or not.

About the Sunflower layout, it can allow either isolated environments, where certain musicians or musical things just interact with the environment, sending and receiving information from it, or collaborative environments, where users exchange data with each other.

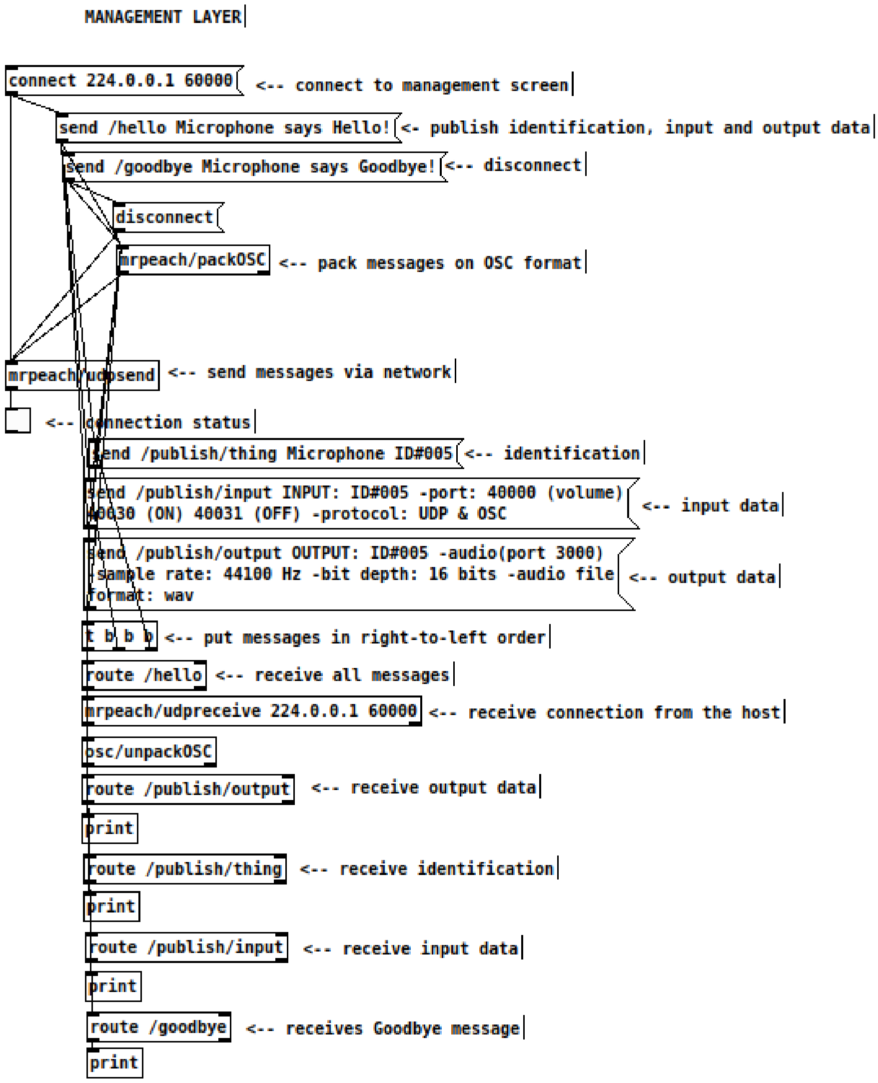

Figure 5. Drum machine patch control.

Source: The Author.

4. Feasible Scenarios

From the elucidation of how Sunflower works, its layers, data categories, and protocols it supports, some usage scenarios may become possible. The following section, therefore, takes care of showing them, portraying a Jam among a group of people, a studio based on the principles of IoMusT, and an artistic presentation focused on the interaction between musicians and audience members.

4.1 Scenario 1 - A Jam Session with Acoustics and Devices

The first scenario is a jam session that combines traditional instruments and electronic devices that exchange information over the network. Traditional instruments are plugged into speakers or patched together, while some users/musicians can participate by patching their smartphones or computers.

The video can come from the cameras of these same cell phones or laptops and send over the network to SmarTV's or screens, which will show them to all participants. Still, images, videos, animations, and other graphic elements can also be used to ensure an audiovisual experience. Controlling other devices, such as triggering a sequence of notes on a synthesizer and changing the pattern and BPM of the drums, can also be performed through video equipment.

Some users can participate by controlling volume, recording, instrument effects, and turning them on or off over the network. They can also change the color patterns of the images, change their resolution, format, etc.

A sound technician can be responsible for the management layer, handling connections, allowing or disallowing certain users to communicate, as well as choosing which instruments and audio tracks will be sent to the public address.

4.2 Scenario 2 - A IoMusT Based Studio

A second possible scenario is a studio that uses IoMusT concepts to record solo artists, duos, and small groups, as well as orchestras with the most varied instruments. For this, the recording interface can adapt its size according to the amount of equipment connected to it. Musicians can record even if they are not in the same physical location, and audio files can be recorded for later mixing and mastering.

The graphic layer can provide technical information about the network, such as which objects are connected and how to connect to them, and also about the recording, such as displaying sheet music, lyrics, and other information that helps in the recording.

Pedal stomps can be reconfigured and controlled remotely or via smartphones. Likewise, the sound engineer can define which channels will be used and combine pre-recorded tracks with live performance. Other musical parameters can be automatically corrected by scripts or artificial intelligence systems. All of this will be done at the control layer.

The management layer will be responsible for connecting the instruments to the recording interfaces, controlling the graphical information that will be displayed, which elements are subject to control, etc.

4.3 Scenario 3 - Live Performances

Like the two possible scenarios mentioned, this one that deals with live presentations is also full of possibilities. Musicians can exchange data over the network instead of traditional audio systems, and audience members can corroborate the musical composition through their own devices.

The graphics layer can be used to convey participation information to the audience, display videos, and animations that expand the artistic capabilities of the performance, and again present some musical aid such as displaying lyrics, setlist, and sheet music to everyone involved.

The control of the guitar pedals, lights, and screens can be done by audience members or by specialized technicians. In the same way, this layer can replace the traditional means of mixing and regulating the sound with digitalized means controlled by the network.

The management layer can restrict access to certain functionality and handle permissions in the environment, whether relating to music or network access. At times, it blends in with the control layer.

5. Discussion

Although there are few and dispersed IoMusT environments, there is already some effort to deal with the possibilities and dilemmas proposed by this area. Therefore, it is important to establish some parallels and highlight the differences between these models and the Sunflower. To do this, we are going to make a comparison with three pre-existing ecosystems [9].

The second model is more focused on expanding the interaction capabilities between musicians and their instrument, and also between musicians and audiences. To do so, it uses a Smart Musical Instrument (SMI), capable of collecting and sending data over the network, and smartphones, capable of searching for audio samples in a repository, enabling the increase of narrative and musical expressiveness. The SMI audio engine is also built on Pd, and it uses UDP and OSC protocols to exchange information [9,13].

The third and last model to be presented is also the most experimental. That's because it intends to solve the reliability and latency problems in IoMusT environments that are implemented in cellular networks using 5G technology as a communication module for musical things. Precisely due to the experimental nature, the authors were not concerned with questions about audio format, music information protocols, and communication in mobile networks [4].

From this brief presentation of the other models, some similarities with the Sunflower can be observed, such as the common use of Pure Data as an audio engine and the prevalence of the OSC and UDP protocols for communication and the client-server architecture. However, some differences are quite clear. Specifically dealing with Sunflower, only it features video and control layers. This is because the authors have thought of this environment not only to meet the musical and communication issues, but also to expand the artistic capabilities of this type of environment. In this way, users can participate in the show using other devices besides music.

Another highlight in this environment is that, unlike his peers, he doesn't relegate to the Pd only the role of sound effects' creator, so he puts this tool at the center of the development of his layers. The main one, as it could not be different, is in the design of the audio layer. Thanks to its ability to be used in live performances and the infinity of possibilities it has, the language was used in the construction of several patches, from those that simulate musical things (such as recorder, drums, and tuning fork) to those that allowed the connection of instruments or sound equipment (such as those that allow the connection of guitars, basses, and microphones to the network).

Another aspect that is still little explored in IoMusT environments is the use of video, largely because of the main discussion about these ecosystems being solely about music creation, in addition to the fact that video data is larger in byte sizes, which requires an infrastructure more robust network. To deal with this in Sunflower, the authors used GEM, a powerful external that allows video capture from any available source and also generates other graphical information. Added to the ability to send data over the Pd network, this layer expanded its concepts beyond the conventional, allowing a better manipulation of visual data in the environment.

Although the Control Layer is centered on the use of the OSC protocol, Pd again plays an important role, either because it natively allows the use of this protocol, not requiring third-party tools for it to be used, or because it has its basic capabilities augmented by the OSC, allowing it to be used to control buttons, sliders, and messages.

The Management Layer, which has a script for complete operation, also finds solutions in Pd, as the tool allows sending the main characteristics of each musical thing to the administrator, facilitating the allocation of resources in the environment.

Overcoming this structural discussion of Sunflower, it is worth highlighting the way it works, inspired by the Pipes-and-Filters architecture, which is very similar to the Pure Data programming model. In both, the elements are autonomous and independent, and when they appear in the environment, they do not know the properties of the other objects present. Thus, it remains for them to establish some type of communication based on the data they receive as input and the data that are sent by their outputs. Because of this, Pd not only acts on all layers of the environment but also inspires, in a way, its way of functioning.

In this way, the contributions of the Pd are clear not only for the elaboration of the Sunflower, being present in all stages of the process and making them easier to be carried out, but also for the emerging area of IoMusT, which, despite being new, presents clear objectives and difficulties, such as the difficulty of dealing with the heterogeneity of their environments, the different levels of musical and technological knowledge that users of these environments have, artistic, environmental and social concerns, among several others that may, in part, be solved from the basic functionalities of the language.

Despite the prevalence of the system's strengths, some negative points can be mentioned, such as the need for some knowledge of Pure Data to develop this environment, as it uses medium or advanced language resources, as well as the need for a patch to connect the network objects, especially those that do not have this characteristic naturally. The authors believe that by transposing some of these features to smartphones, which can be considered ubiquitous devices, some of these usability issues can be resolved.

6. Conclusions

The present study presented an IoMusT environment centered on the Pure Data figure, proposing a counterpoint to those previously developed, because the model presented here uses this tool not only as an audio engine but also to generate graphical data, to allow control of musical things remotely, send data to the network and also to help in managing the environment. In addition, its mode of operation is inspired by the Pipes-and-Filters architecture, which allows musical things without prior knowledge of their neighbors to communicate. A categorization of these objects inspired by this architecture was also presented.

When thinking about IoMusT ecosystems, especially those related to maker culture, some elements can be thought of to assist in music creation, while they can also deal with issues related to the network, such as the ESP8266 and ESP 32 microcontrollers, given their ability to connect to the network wirelessly. However, as they do not have an operating system, they cannot have other tools built into them, in particular Pure Data. From this, arises the provocation to allow these devices to somehow exchange information with the Pd. The same is true for buttons and sliders that work in a standalone way and that can help these environments, as they are able to exercise some communication.

Another important point to highlight about the functioning of musical things created in Pd is the ability to modify their behavior since the codes can be updated to suit the environment and other musical things present in it.

Modeling the environment in this way opens up new possibilities for collective artistic creations, such as live coding, where not only the audio will be modified in real-time based on the programming, but also the structure of the objects involved; the laptop and mobile orchestras, which will be able to count not only with several instruments but also with graphical and management functionalities that will facilitate the execution of the piece; and installations and presentations, which will become more accessible in financial and computational terms, as well as having expanded their artistic and participatory capacities.

The authors, by no means, want to end the discussion about IoMusT environments or about the use of Pd in the most varied musical activities, they intend to demonstrate a new way of thinking about these ecosystems using the technical features of Pure Data that go beyond making artistic.

Acknowledgments

Authors would like to thanks to all ALICE members. The authors would like also to thank the support of the funding agencies CNPq (Grant Number 151975/2019-1), CAPES (Grant Number 88887.486097/2020-00) and FAPEMIG.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] Burges, R.J., 2014. History of Music Production. Oxford, UK: Oxford University Press.

[2] Vieira, R., Schiavoni, F., 2021. Sunflower: an environment for standardized communication of IoMusT. Audio Mostly 2021. Association for Computing Machinery, New York, NY, USA. pp. 175-181. DOI: https://doi.org/10.1145/3478384.3478414.

[3] Millard, A., 2005. America on Record: A History of Recorded Sound. Cambridge, UK: Cambridge University Press.

[4] Centenaro, M., Casari, P., Turchet, L., 2020. Towards a 5G Communication Architecture for the Internet of Musical Things. Conference of Open Innovations Association, Trento, Italy. pp. 38-45.

[5] Atzori, L., Iera, A., Morabito, G., 2010. The Internet of Things: A survey. Computer Networks. 10, 2787-2805.

[6] Turchet, L., McPherson, A., Fischione, C., 2016. Smart Instruments: Towards an Ecosystem of Interoperable Devices connecting Performers and Audiences. IEEE Open Innovations Association.

[7] Turchet, L., Fischione, C., Essi, G., Keller, D., Barthet, M., 2018. Internet of Musical Things: Vision and Challenges. IEEE Access. 06, 61994-62017.

[8] Vieira, R., Gonçalves, L., Schiavoni, F., 2020. The Things of the Internet of Musical Things: Defining the Difficulties to Standardize the Behavior of These Devices. X Brazilian Symposium on Computing Systems Engineering, Online. pp. 1-7.

[9] Turchet, L., Fischione, C., Essi, G., Keller, D., Barthet, M., 2018. Internet of Musical Things: Vision and Challenges. IEEE Access. 06, 61994-62017.

[10] Turchet, L., Antoniazzi, F., Viola, F., Giunchiglia, F., Fazekas, G., 2020. The Internet of Musical Things Ontology. SSRN Electronic Journal. 01.

[11] Vieira, R., Schiavoni, F., 2020. Can Pipes-and-Filters architecture hel creativity in Music?. X Workshop on Ubiquitous Music, Online.

[12] Bijlsma, A., Roubtsova, E., Stuurman, S., 2011. Software Architecture. Catalunya, Spain: Free Technology Academy (FTA).

[13] Turchet, L., Barthet, M., 2017. Envisioning smart musical haptic wearables to enhance performers' creative communication. International Symposium on Computer Music Multidisciplinary Research.