Web Semantics: Science, Services and Agents on the World Wide Web 60 (2020) 100548

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

The Internet of Musical Things Ontology

Luca Turchet [a,*], Francesco Antoniazzi [b], Fabio Viola [b], Fausto Giunchiglia [a], György Fazekas [c]

[a] Department of Information Engineering and Computer Science, University of Trento, Italy
[b] Advanced Research Center on Electronic System, University of Bologna, Italy
[c] Center for Digital Music, Queen Mary University of London, United Kingdom of Great Britain and Northern Ireland

ARTICLE INFO

Article history:
Received 29 July 2019
Received in revised form 22 October 2019
Accepted 14 January 2020
Available online 22 January 2020

MSC:
00-01
99-00

Keywords:
Internet of Musical Things
Smart musical instruments
Semantic audio

ABSTRACT

The Internet of Musical Things (IoMusT) is an emerging research area consisting of the extension of the Internet of Things paradigm to the music domain. Interoperability represents a central issue within this domain, where heterogeneous objects dedicated to the production and/or reception of musical content (Musical Things) are envisioned to communicate between each other. This paper proposes an ontology for the representation of the knowledge related to IoMusT ecosystems to facilitate interoperability between Musical Things. There was no previous comprehensive data model for the IoMusT domain, however the new ontology relates to existing ontologies, including the SOSA Ontology for the representation of sensors and actuators and the Music Ontology focusing on the production and consumption of music. This paper documents the design of the ontology and its evaluation with respect to specific requirements gathered from an extensive literature review, which was based on scenarios involving IoMusT stakeholders, such as performers and audience members. The IoMusT Ontology can be accessed at: https://w3id.org/iomust#.

1. Introduction

The Internet of Musical Things (IoMusT) is an emerging research area consisting of the extension of the Internet of Things paradigm to the musical domain. This field is positioned at the confluence of music technology, the Internet of Things, human–computer interaction, and artificial intelligence. IoMusT relates to the networks of computing devices embedded in physical objects (Musical Things) dedicated to the production and/or reception of musical content. Considering the computer science perspective, Turchet and colleagues defined a Musical Thing as *"a computing device capable of sensing, acquiring, processing, or actuating, and exchanging data serving a musical purpose"*. The IoMusT was then defined as *"the ensemble of interfaces, protocols and representations of music-related information that enable services and applications serving a musical purpose based on interactions between humans and Musical Things or between Musical Things themselves, in physical and/or digital realms. Music-related information refers to data sensed and processed by a Musical Thing, and/or exchanged with a human or with another Musical Thing"* [1].

Various kinds of Musical Things can be envisioned, which may be categorized according to the musical purpose they serve (e.g., to control, generate, or track responses to musical content). Examples of existing Musical Things are the "smart musical instruments", a new family of musical instruments encompassing sensors, actuators, wireless connectivity and on-board processing [2]. These musical devices are able to directly exchange musically-relevant information with one another as well as communicate with a diverse network of external devices, such as smartphones, wearables, virtual reality headsets or stage equipment. Instances of smart musical instruments include the Smart Cajón reported in [3] and the Sensus Smart Guitar developed by MIND Music Labs [4]. Another example of Musical Things are "musical haptic wearables" [5–7], a novel class of wearable devices embedding haptic stimulation, tracking of gestures and physiological parameters and wireless connectivity features. On the one hand, such devices were conceived to enhance communication between performers as well as between performers and audience members by leveraging the sense of touch in both co-located and remote settings. On the other hand, they were devised to enrich musical experiences of audiences of music performances by integrating haptic stimulations, as well as provide new capabilities for creative participation thanks to embedded sensor interfaces.

Musical Things are connected by an infrastructure that enables multidirectional communication, both locally and remotely, between different stakeholders such as composers, performers, audience members, audio producers, live sound engineers, as

* Corresponding author.
E-mail address: luca.turchet@unitn.it (L. Turchet).

https://doi.org/10.1016/j.websem.2020.100548
1570-8268/

well as music students and music teachers. The ecosystems that will form around Internet of Musical Things technologies are envisioned to support novel forms of interactions between such stakeholders by means of novel musical applications and services. This has the potential to revolutionize the way music is composed, performed, experienced, learned and recorded.

To accomplish the IoMusT vision, the Musical Things within an ecosystem need to communicate through a common language. A central unsolved issue within the IoMusT paradigm is how facilitating interoperability among heterogeneous Musical Things, which may serve radically different purposes (e.g., real-time analysis of musical content, generation and delivery of haptic, visual, or olfactory sensory layers additional to the musical content, delivery of content-recommendation services for music students). To date, interoperability across musical devices has mostly relied on protocols for the exchange of musical messages such as Musical Instrument Digital Interface (MIDI) or Open Sound Control (OSC) [8] and tools based on it (e.g., libmapper [9]).

Today, MIDI is a widely adopted protocol, which was conceived in the 80 s to enable the exchange of information across musical instruments developed by different vendors. MIDI is not well suited to achieve interoperability across heterogeneous devices because it was specifically conceived for communication across musical instruments. Moreover, MIDI is very limited in resolution (e.g., it uses integers between 0 and 127), which prevents to represent information with a high level of detail and accuracy. On the other hand, OSC is a protocol more flexible and with higher resolution than MIDI, as it enables user-defined namespaces and supports messages with various formats (including floats and strings). This would make OSC more suitable to facilitate communication across heterogeneous devices, including non-musical ones. However, OSC is not equipped with standard namespaces for interfacing devices and as a consequence, connected devices neither know each other or each other's capabilities. Tools based on OSC, such as libmapper and Sense-World DataNetwork, have been proposed where a semantic layer is added to the conventional OSC protocol structure [10–12]. These protocols provide decentralized resource allocation and discovery, and flexible connectivity letting devices describe themselves and their capabilities. However, they target the use of a LAN subnet where support for multicast can be guaranteed [9]. They are not conceived for Web-based interactions nor they support mechanisms of automatic inference.

In summary, the existing musical protocols are not adequate to support interoperability across the wide heterogeneity of Musical Things, as they are typically not flexible, lack high resolution, not equipped with inference mechanisms, and do not support easy integration with the Web. Semantic technologies, such as semantic web [13] and knowledge representation [14], possess these features. For this reason, they have been recently envisioned as a solution to enable interoperability across heterogeneous Musical Things [1]. Existing ontologies devised for the musical domain to date, such as the Music Ontology [15], the Studio Ontology [16] or the Audio Features Ontology [17], are insufficient to represent the wide knowledge base that the variety of the possible Musical Things entail. An ontology specific to the IoMusT scenario is currently missing. As a consequence, the use of semantic technologies in Internet of Musical Things contexts is limited to scenarios involving homogeneous Musical Things serving similar musical purposes or ad-hoc interactions designed for a specific, fixed scenario.

The first effort towards the application of semantic technologies to the IoMusT context is reported in [18]. The authors proposed a semantically-enriched Internet of Musical Things architecture relying on a semantic audio server and edge computing techniques. Specifically, a SPARQL Event Processing Architecture [19] was employed as an interoperability enabler allowing multiple Musical Things to cooperate, relying on a music-related ontology [17]. A limitation of the developed architecture was the involvement of an ontology restricted to the representation of simple musical features, which prevented Musical Things dedicated to purposes other than music generation to join the ecosystem formed around the architecture.

Semantic technologies based on an ontology for the IoMusT can assist in managing, querying, and combining information characterizing an IoMusT ecosystem, including data about the music produced, the involved stakeholders, the utilized Musical Things and their application and services. This has the potential to spur the exploration of novel artistic avenues, such as performance and composition, for instance based on emergent properties of an IoMusT ecosystem [20].

In this paper, we propose the "Internet of Musical Things Ontology", an ontology devised to represent the knowledge related to IoMusT ecosystems. We describe the design process of the IoMusT Ontology and its first (and current) version, i.e., 1.0.0. The description of the IoMusT Ontology follows the MIRO (minimum information for the reporting of an ontology) guidelines [21]. For reference, the paper reports the MIRO designations (e.g., E.9 for Ontology relationships), where the specific information item is provided. The ontology name (A.1) and its need (B.1) have been already introduced. The ontology is available at https://w3id.org/iomust# (A.4) with license GPL3 (A.3).

2. Methodology, audience, and scope

This section describes the methodology adopted for the design and development of the IoMusT Ontology, as well as the audience of the ontology and its scope.

2.1. Methodology for ontology development

The ontology is developed and maintained by the authors and other members of the emerging IoMusT research community, which is currently composed by leading research institutes in sound music computing and Internet of Things (A.2 and C.2).

The design and development of the IoMusT Ontology was mostly inspired by the METHONTOLOGY methodological framework [22] (A.6). This framework is composed of six phases: (i) the specification, i.e., the identification of the audience, scope, scenarios of use, and requirements (Sections 5 and 2.2); (ii) the conceptualization of an informal model (first paragraph of Section 6); (iii) the formalization of the ontology namespaces, classes and properties; and (iv) the integration of existing ontologies in a description and its formalization and publication using OWL2 [23] (Section 6); (v) the implementation of the ontology with an appropriate serialization language (Section 7); (vi) the maintenance of the ontology once implemented (Section 7).

Moreover, the METHONTOLOGY framework identifies three tasks that are accomplished during the lifetime of the ontology. These are orthogonal to the five phases: (i) knowledge acquisition through research of related ontologies and models (Section 3) as well as gathering data from potential users (Section 4), to inform multiple phases of the design process, mainly conceptualization and integration; (ii) documentation of the process phases (internal) and the ontology specification (public) (Section 7); (iii) the evaluation of the ontology before its release (Section 8).

Other works, like Uschold [24] and more recently De Nicola and Missikoff [25] suggest different methodologies for ontology engineering. These approaches, however, focus on techniques to formalize new ontologies from scratch. This is not the case in the current research, where the goal is to provide a new contribution based as much as possible on the integration of pre-existing ontologies.

2.2. Scope and audience

The role of the IoMusT Ontology is to offer a common data model enabling interoperability among heterogeneous Musical Things, which allows both people and virtual agents to seamlessly generate, explore, access, or transform music-related content produced within an IoMusT ecosystem. Therefore, the scope of the ontology (C.1) is represented by all ecosystems forming around existing or future IoMusT technologies.

The target audience of the ontology (B.3) is represented by all actors and stakeholders that are involved in such ecosystems, including performers, audience members, composers, studio producers, live sound engineers, and choreographers.

3. Related ontologies and data models

Before defining an ontology specific to the IoMusT domain we conducted a review of existing ontologies. The review showed that no existing ontology covers the requirements of the identified use cases or satisfied a design goal of representing concepts and relations in the context of networked musical activities (see Sections 4 and 5). The IoMusT vision is intrinsically multisensory and highly interdisciplinary [1]. This section describes ontologies and data models (B.2) that are related to such a vision. They have been gathered through the research of literature and online resources (D.1 and D.2) and evaluated as part of the design process (D.3).

3.1. Ontologies for the audio domain

Several ontologies have been proposed in recent years for the audio and music domains, in recognition of the complexity and broad ranging applications of such ontologies, and the fact that much of the information exchanged on the Web today is multimedia, of which music is a very substantial component, rather than text. The scope of such ontologies are wide ranging, starting from very focused areas of music production such as audio effects [26], to larger binding ontologies that target the description and retrieval of audio resources on the web in general [27].

The Music Ontology (MO) [15,28] is a general purpose high-level ontology for the music domain that models the music value-chain from production to consumption. Therefore its focus is on editorial metadata, e.g. artist name and title associated with audio recordings, as well as the representation of major steps in the production of recorded music, from composition, through performance and recording, to release. The ontology is in sharp contrast with music metadata standards that typically suffer from the limitations of an object or item centric view [29] and unique in its ability to cover the four major stages in the development of intellectual works identified by the library information science community in the context of Functional Requirements of Bibliographic Records (FRBR) [30]. To this end, MO relies on and extends the model in [31], and provides an event based conceptualization of music production workflows. Therefore it deals with the notion of *musical works*, *expressions*, *manifestations* and *musical items*, to identify e.g. a `mo:MusicalWork` and its performances by different artists, and potentially different recordings and releases. MO binds these concept together using events from the Event Ontology,[1] to describe transitions between states of intellectual works. For instance, placing a microphone in front of an instrument implies a recording event (`event:Event`) that facilitates transition from one representation of a work to another (*sound* to *signal*). When the recorded signal is transferred to a medium and released (a release event) we move from *musical expression* to *musical manifestation*. Numerous extensions of the ontology exists, including music theoretical domains such as chords and musical temperament.

The Music Ontology does not deal with the nuances of technical workflows in music production. This is the area covered by the Studio Ontology (SO) [16]. The ontology uses hooks provided by MO to describe what happens between the *expression* and *manifestation* layers. This includes typical procedures in audio engineering as well as signal processing. The Studio Ontology covers for instance, microphone placement, physical signal connectivity (e.g. studio wiring), mixing, editing and mastering of audio, a process involving several sound signal transformations. The core model and innovation of this ontology is a parallel *event flow* and *signal flow*, aiming to describe a series of actions performed by audio engineers, coupled with a series of signal transformations together with the technical artifacts involved in them and their configuration parameters. This allows for instance to trace signal transformations using the model described in [32]. From a more philosophical perspective, this ontology allows for capturing the contribution of the producer and engineer to a music piece, which, at least in modern music, is just as important as the composition.

The Audio Features Ontology (AFO) addresses another audio domain that requires detailed conceptualizations. Audio Features are descriptors that represent specific characteristics of sound signals. These may relate to measurable properties of the signal, such as bandwidth or spectral centroid, perceptual qualities like pitch and loudness, and musical characteristics such as notes, musical key and chords. A benefit of smart instruments is their ability to process audio and extract features that are relevant in a particular interaction scenario. Therefore a formal model of audio features is crucial to provide interoperability in IoMusT. The AFO defines a four layer conceptual model, akin to FRBR, to describe audio feature types, coupled with their mathematical or computational models, implementations and outputs. The associated audio feature vocabulary (AFV[2]) defines a large number of audio descriptors.

The Musical Instruments Ontology [33] is highly relevant to the domain of IoMusT. It provides an ontological model for encoding well known instrument classification systems, e.g. for grouping instruments into categories such as *Idiophones* or *Aerophones*, based on their sound production or excitation mechanism. The ontology proposes a solution to deal with terminological heterogeneity among different knowledge representation systems in this domain.

The Audio Commons Ontology (ACO) [27] is an example of a higher level domain ontology that binds several audio related ontologies together. It was designed to facilitate the integration of audio content repositories on the Web as well as content consumption by software agents. The goal of this ontology is to facilitate the description and retrieval of audio content in professional content production. It follows the layered conceptualization paradigm introduced in MO and FRBR, but extends this to non-musical sounds. ACO also deals with professional requirements for sound metadata (such as those of broadcasters) by integrating the EBU Core ontology.[3] For IoMusT, this provides an example of an ontology that supports integration across heterogeneous domains.

The Context-based Music Recommendation (COMUS) ontology [34] is aimed at modeling user's musical preferences and context. It supports reasoning on the user's desired emotions and preferences. As the name suggests, the COMUS ontology is mostly oriented at providing music recommendations.

1 http://purl.org/NET/c4dm/event.owl#.

2 https://semantic-audio.github.io/afv/#.

3 https://www.ebu.ch/metadata/ontologies/ebucore/.

3.2. Ontologies for the haptic domain

Ontologies for the haptic domain are also important, due to the fact that musical haptic wearables [5] and other devices in the IoMusT environment often require tactile interaction. Myrgioti et al. in [35] proposed a model for tactile information flow and an ontology developed using UML. This work has then been refined through the adoption of OWL and in-depth analysis of the process of developing haptics software [36]. Adhami and El Saddik in [37] present the Service Oriented Development of Haptics Ontology (SODHO). It is aimed at a unified modeling of sensors and actuators (or in general transducers) for haptics and it is mostly based on HASM [36].

Albert et al. [38] focused on tactile perception by analogy with visual perception. The authors proposed a generic analysis of the possible ways to represent tactile sensations, and went further by considering existing ontologies that can be somehow related to human perception (e.g., by means of sensors and observations). First, they focused on the SSN ontology [39] that introduces the stimulus–sensor–observation pattern, necessary for the tactile perception.

3.3. Ontologies for sensors, actuators, and connectivity

Among the ontologies designed for the IoT, two of the most diffuse are SSN[4] (Semantic Sensor Network) and SOSA[5] (Sensor, Observation, Sample, and Actuator). Both SSN and SOSA adopt a complex approach to description of hardware, observation of physical entities and actuation. SSN [39] covers the majority of the SensorML standard[6] and has been designed to describe sensors and observations, as well as the deployment in which sensors are employed. SOSA [40] adopts a lightweight approach to describe sensors, actuators and the acts of observation and actuation. SOSA acts as a replacement of the Sensor–Stimulus–Observation (SSO) design pattern provided by SSN, that provides greater expressivity [41].

At a higher level of abstraction, things in IoT can be represented according to the Web Thing model[7] proposed by the W3C. In this sense, devices are provided with the so-called thing description, a detailed profile reporting properties, events and actions exposed through its interface. A first attempt to semantically represent this model has been provided by Charpenay et al. in [42], later on envisioned by Serena et al. in [43] for a discovery framework. The Web of Things ontology discussed by Charpenay et al. and Serena et al. has been employed by Viola et al. [44] to build a semantic Web of Things environment for recommendations in the audio domain. Eventually, Antoniazzi and Viola [45] provided a semantic version of the Web of Things.

4. Knowledge acquisition

Knowledge acquisition is an activity that has been performed since the initial phases of the ontology building. It is typically continuously carried out. For the purposes of the proposed ontology, we conducted a review of the existing IoMusT literature and related areas. In particular, the studies reported in [1,2,7,46,47] and [5] represent the most relevant sources for the creation of the knowledge base. Such works are based on extensive literature reviews of IoMusT-related topics and contain descriptions of different scenarios involving Musical Things and stakeholders within various IoMusT ecosystems. Moreover, we considered works reporting various experiments with different stakeholders, such as performers and audience members, including co-design of smart musical instruments and related interactions (e.g., [7,48]). Furthermore, we defined additional scenarios consisting of use cases for IoMusT ecosystems not present in the previous literature. Hereinafter, we summarize three instances of scenarios, which represent the most relevant examples of use cases around which the ontology design is framed:

Multisensory live music experience and audience participation. During a live concert the smart instruments of the band deliver to the audience's Musical Things some messages that control in real-time their behavior, i.e. the Musical Things respond to those messages by delivering a sensory feedback augmenting the auditory content (e.g., visualizations provided by smart glasses, vibrations produced by jackets enhanced with vibro-tactile motors). The delivered messages, and as a consequence the provided sensory feedback, depend on features of the music played, which are extracted by each instrument. The same Musical Things can track the movements of the audience (e.g., detecting when they are dancing), and this information is used to produce a musical accompaniment for the band. At certain points of the concert, also the remote audience, which is connected to the concert venue, participates in the music creation process by means of gestures on their smartphones, following instructions that are displayed on them and delivered by the smart instruments. These instructions depend on the geographical position of the remote audience.

Enhanced music learning. During her practice activities, a guitar student uses a smartphone app connected to her smart guitar. The smart guitar detects the errors made by the student and the smartphone app provides recommendation about how to improve her playing and which musical piece to play next. Such recommendations are based on a connected cloud-based service that receives information on the act of playing of the student, which is retrieved by the smart guitar. Following the recommendation service's suggestions, the student accepts to play the suggested musical piece. This choice automatically configure the smart guitar with the effects chain needed to practice that piece.

Remote rehearsals and intelligent music productions. Two musicians use their smart instruments to rehearse together at a distance of 100 km, thanks to the direct connectivity between their instruments. At some point they make a recording of their music, which also encompasses metadata related to the configuration of the sound engine of the smart instruments, where each sensor in the sensor interface is associated to a musical parameter of a certain audio effect (e.g., the values of a pressure sensor are mapped to the feedback parameter of a delay effect). Such multi-layer recording is sent to studio producers who use all the available information first to recreate an authentic rendering of the rehearsal in their studio environment, and then to create a new version of the recording (e.g., by modifying the mapping function between a sensor and a parameter of the associated audio effect).

5. Specification

The acquired knowledge was then analyzed to identify a set of requirements that the ontology should satisfy [49]. The literature review led to a total of 15 scenarios (5 scenarios from [1], 5 from [2], and 5 defined by the authors or derived from recent experiments with users described in the literature). For each scenario we derived a set of requirements, and then applied a thematic analysis [50] to reduced them. The resulting requirements are represented below as a list of example questions that the ontology should be able to support answering [51], and a list of formal requirements.

4 https://www.w3.org/TR/vocab-ssn/.

5 https://www.w3.org/2015/spatial/wiki/SOSA_Ontology.

6 https://www.opengeospatial.org/standards/sensorml.

7 https://www.w3.org/Submission/wot-model/.

5.1. Competency questions

The following sample questions are meant to be asked with respect to an IoMusT ecosystem:

1. Which type of Musical Things are used by the local and remote performers during the live concert?
2. How many Musical Things used by the audience provide haptic feedback?
3. What smart instruments are controlling the smartphones used by the audience?
4. What is the mood of the music at a given time during the live performance?
5. How many audience members are actively participating in the music creation process thanks to their Musical Things?
6. Which kind of stage equipment is used at a given time during the concert?
7. Which gestural and biometric parameters are tracked from the audience during the live performance?
8. How many and which kind of networks are used during a performance?
9. What pedagogical applications are available for a smart violin?
10. With which music content repository a smart ukulele can interact?
11. Which services are available for a smart guitar and what are their purposes?
12. What type of sensors and actuators compose a smart musical instrument or a musical haptic wearable?

5.2. Formal requirements

The IoMusT Ontology should be able to:

1. represent the concept of Musical Things, including:
   (a) its type (e.g., musical instrument, wearable device, stage equipment);
   (b) its characteristics including the number and type of inputs (e.g., sensors tracking movements or biometric parameters) and outputs (e.g., auditory, visual, haptic, olfactory);
   (c) the type of person for which it is conceived (e.g., performer, audience member, live sound engineer, producer);
   (d) its function (e.g., a smart instrument used to produce musical content, a musical haptic wearable aiming at enriching the listeners' musical experience, an interface used by audience members for participatory purposes, a device used to infer the mood of audience members based on sensed quantities);
   (e) its geographical position;
   (f) the type of data that it generates (e.g., audio signal, text message);
2. represent the concept of connectivity, including:
   (a) the type of network involved (e.g., local network, remote network, Wi-Fi-based, millimeter waves-based);
   (b) the attributes of the network (e.g., bandwidth, speed, synchronization mechanisms);
   (c) the time taken by the network to deliver/receive a message to/from a certain Musical Thing;
3. represent the concept of application and service, including:
   (a) its purpose (e.g., for music learning, performance, composition, studio production)
   (b) its level of interactivity (e.g., interactive, non-interactive)
   (c) its type (e.g., social network, online music content repository)
   (d) its user (e.g., composer, performer, studio producer, educationalist, student, audience member)
4. describe attributes of the music (produced live) at a given time, including:
   (a) low-level features (e.g., the density of notes);
   (b) high-level features (e.g., the mood)
5. describe attributes of the ecosystems, including:
   (a) the number and type of Musical Things present in the network at a given time and a given space;
   (b) which Musical Things are interacting;
   (c) the number and type of applications and services available within the ecosystem;
   (d) the number and type of networks used at a given time.

6. Ontology description

The Internet of Musical Things ontology (the IoMusT Ontology) has been developed incrementally. It is well understood that the task of developing ontologies is in general complex and it requires an approach that involves continuous refinement and checking of concepts and relationships. This can be done in several ways, and may be performed iteratively as long as the expected match of the ontology with the real subject or knowledge domain is achieved.

Not surprisingly, the first step is to split the domain of interest in smaller parts if possible. For each of those smaller parts, secondly, iterations are needed to ensure that all relevant concepts are included. Sometimes this is done by surveying a pool of experts and/or future users of the ontology, to obtain their feedback. Clearly, this check helps designers to avoid wrong naming on resources, as well as to detect and correct contradictory assertions.

Then, the smaller parts have to be joined together to form the ontology. Again, the expressiveness of the complete work has to be checked, and in this paper this is provided by requirement analysis and evaluation. The question to be answered, here, is: *is my ontology capable to describe my context? If so, is the description made with the precision needed?* This process also may be performed iteratively.

The IoMusT Ontology is not an exception. On the contrary, it is very important to notice that the formalization of a vocabulary for the Internet of Musical Things requires this feedback process to ensure a coherent representation of music-related entities with general-purpose contents.

A bottom-up process was deployed for our case. In particular, jumping from wider to narrower concepts, the first idea to be discussed is indeed the connection that stands between the global interpretation of Internet of Things and how to subsume it into the Internet of Musical Things. See also Fig. 3. Clearly, the former is larger than the latter, which should represent a specialization and rely on it. The usage of the IoMusT Ontology, as a consequence, should allow a transparent view of any Musical Thing context as an IoT system. There would be no point in ignoring this core aspect, because the core idea of ontology engineering is to provide a shared and interoperable way to collaborate between different fields of knowledge. Any design choice opposed to this view would have as a direct result the creation of another vertical silos within the IoT chaos [52].

In order to replicate in the ontology this necessary duality, this work will suggest the adoption of two new namespaces:

1. iot, that will be used to connect concepts that belong to the broader view of generic devices;
2. iomust, which is an extension of iot defined as iot:musical. Within this namespace are organized the concepts of music-related IoT;

For the sake of clarity, in the present paper the prefixes are kept in a contracted form. To see their expanded version, please see Table A.3 in Appendix.

6.1. iot namespace

The first concept to be defined is the *Thing* as it is intended in the acronym IoT. Liu and Baiocchi [53] survey and comment the spectrum of definitions that have been suggested in literature over the time. Among the surveyed entries, the one proposed by the IEEE is coherent with our requirement of generality: the thing "is any physical object relevant from a user or application perspective", meaning that we consider things as items exploiting, or being exploited by, other items. Therefore, from now on, the class iot:Thing has to be considered according to this definition. Notice that also regular everyday life objects may be iot:Things, like chairs, pillows, a scarf, a painting and, indeed, a musical instrument.

Clearly, this is a generic class that needs to be further specialized in subclasses. Again, substantial help may come from the listings in [53], as iot:Thing is definitely a large container. For instance, things can be wearable objects: so, the class iot:WearableThing can be defined to represent this category. Similarly, devices can also be *smart*, so we call for iot:Smart Thing class: smart things, e.g., a smartphone, a smart TV, include special technological features or artifacts that provide them with relevant added value over the basic version of the same object.

Eventually, things can be connected to a communication network: they are, in this case, instances of the class iot:Connected Thing. Notice that the aforementioned Studio Ontology [16] contains a rich environment of properties and classes related to connectivity (e.g., the *Connectivity* and the *Device* sub-ontologies).

It would neither be reasonable, nor useful, to list here a hundred of possible subclasses. For this reason, in the present paper only a few will be defined, as the discussion requires them. It is important to notice that there is complete freedom to include new classes whenever needed, as this is precisely the kind of incremental approach for ontology engineering which was mentioned above.

6.2. Musical things in the iomust namespace

The IoMusT Ontology, as already discussed, aims to develop the iot namespace in its musical flavor. To do so, the reference to a vocabulary connected to music is essential. Our reference in this work was introduced in Section 3.1: it is the Music Ontology [15], which will be mentioned as the music namespace. An important contribution of this namespace in IoMusT Ontology is its supporting role in creating the archetype of Musical Thing, i.e. the class iomust:MusicalThing. In the present work, our definition for this class is the following: the Musical Thing is *a thing used to produce or enjoy music, with reference to its context*. As a consequence, IoMusT Ontology will consider that a smart loudspeaker, or a CD by David Bowie belong to that class, as well as a smart violin located in a concert hall. The same smart violin, however, if stored for exposition in a museum, is no more an iomust:MusicalThing because it loses its musical production interest.

The class iomust:MusicalThing is indeed less generic than its superclass iot:Thing, because it provides a light form of contextualization. Yet, however, we need more precise solutions to be even less abstract. All the items identified in the example above (the smart loudspeaker, the CD, the smart violin) would point to iot:Thing through the rdf:type predicate. Then, to include an explicit reference to music, and introduce the Internet of Musical Things namespace, the following rule applies:

Rule 1. If an iot:Thing instance is also connected through rdf:type to a class belonging to the Music Ontology, then it is also an instance of iomust:MusicalThing.

A typical application of Rule 1 is the aforementioned smart violin: consider Listing 1 as an example, where a simple triple representation is given of the implication expressed. Notice that Rule 1 is not intended to be strictly reversible: during a concert, lights and smoke machines may be intended as Musical Things because of their essential contribution to the listening experience, and yet may not be included in one of the music namespace categories.

Listing 1: Triple representation of Rule 1. Extended prefixes are available in Table A.3.

```
ns:SmartViolin a iot:Thing, mo:Instrument
⇒
ns:SmartViolin a iomust:MusicalThing
```

The Generic Musical Thing definition is not enough to build the complete IoMusT. In the following, a sequence of new classes are introduced in the iomust environment, descending from Musical Thing. Each of the classes here correspond to a rule similar to Rule 1 in the OWL.

iomust:SmartMusicalThing is a Musical Thing that is also an iot:SmartThing;

iomust:SmartInstrument is a Musical Thing that is also a mo:Instrument;

iomust:WearableMusicalThing is a Musical Thing that is also an iot:WearableThing;

iomust:StageEquipment is a collection of Musical Things serving as equipment. The definition of collection can be extracted from external ontologies designed *ad hoc* for this, like the one suggested by Ciccarese and Peroni [54].

Table 1 contains some practical examples of usage for the iot and iomust namespace entities.

6.3. iot & iomust sensing, actuating and interacting

So far the discussion on the IoMusT Ontology was conducted as a set of broad definitions for the baseline concepts. Here, instead, space is given to how the integration of other ontologies enables our vision of the IoMusT from a lower level standpoint.

First of all it is necessary to describe the smart devices more in detail, and include additional information related to the electronic devices embedded in the iot:Thing (e.g., micro-controllers, sensors, actuators). The iot:SmartThing was previously introduced to this effect, though without any other specificity. Consequently, to provide greater precision on the actual available sensing and actuating units, other information is needed. Taking into consideration Table 1 as an example, we have to provide a way to semantically distinguish between two instances of iot:SmartThing, like the smart violin, and the virtual reality headset, based on their setup. To achieve such goal, this work suggests the inclusion of an ontology already existing and well known in the panorama, namely, SOSA. The choice of SOSA has three main advantages that greatly benefit the IoMusT Ontology:

Table 1
Example of usage for `iot` and `iomust` namespaces. We here show how objects part of an Internet of (Musical) Things environment can be considered instances of the classes introduced in this research. Extended prefixes are available in Table A.3.

	ns:Bob	ns:Wardrobe	ns:SmartCar	ns:Violin	ns:SmartViolin	ns:TShirt	ns:StageLight	ns:VR_HeadSet	ns:HeartBeatSensor for music experiment
foaf:Person	✓								
iot:Thing		✓	✓	✓	✓	✓	✓	✓	✓
iot:SmartThing			✓		✓		✓	✓	
iot:ConnectedThing					✓		✓	✓	✓
iot:WearableThing						✓		✓	✓
mo:Instrument				✓	✓				
iomust:MusicalThing				✓	✓		✓		✓
iomust:SmartMusicalThing					✓		✓		
iomust:SmartInstrument					✓				
iomust:StageEquipment item							✓		
iomust:WearableMusicalThing									✓

(i) SOSA is *de facto* a light version of SSN, and therefore the IoMusT Ontology can be further extended towards SSN integration very easily; (ii) SOSA is very simple, which is always a relevant factor when studying, building and integrating ontologies; (iii) SSN and SOSA, eventually, are a relatively recent W3C recommendation (the last draft dates back to 2017), which means that they are globally accepted as a reference.

The realization of this ontological alignment is made by including as a plug-in the concept of `sosa:Platform` in the IoMusT Ontology subgraph for the `iot:Thing` and its aforementioned subclasses. According to SOSA documentation, the `sosa:Platform` is an *entity that hosts other entities, particularly Sensors, Actuators, Samplers, and other Platforms*, that is precisely the facet that was missing until now in the `iot` namespace. In Fig. 1 a few examples are provided to show how the connection can be made. As it can be seen, the smart guitar instance `ns:SmartGuitar` has also as `rdf:type` the `sosa:Platform` class. This additional type allows us to include references to the sensors and actuators on board, as well as the entity they measure. Further details on sensing and measurement description, extensively discussed in previous researches like [55,56] and surveyed in [57], are out of the scope of this paper. For the future, anyway, the possibility to integrate new ontologies still exists: for the ones exploiting SOSA and SSN, such process should be trivial.

Sensing and actuating are in general part of a greater intent of interactive IoT system design. Data collection, then, provides the tools to create a feedback to control actuation and, eventually, to show smart behavior. Interaction is an unavoidable part of this process and, consequently, it should also be represented in the ontology alongside with sensors and actuators. Once such semantic prototype is given, it is possible to distinguish the active resources from the environmental passive ones and an interaction is finally possible. Besides, if the semantic view is shared among various systems horizontally, a strong and effective interoperability is automatically achieved.

The study of entities interacting within their environment is a well established field in literature, leveraging the concepts of *agent* (e.g., [59–61] and many others) and *semantic agent* (see, for instance, [62,63]).

Within the IoMusT Ontology, the agent is referred to as any entity, human, object or virtual, that is capable of triggering any kind of dynamic evolution in an environment populated by instances of `iot:Thing` class. Both `iot` and `iomust` namespaces do not include directly such content, as their focus is the device, regardless of the interaction aspect. For this reason, and for the discussion above, the IoMusT Ontology needs to rely on external ontologies to properly provide a definition of agent. Similarly to what has been suggested in the previous paragraphs with SOSA, we suggest here to exploit well-known ontologies, namely FOAF[8] and PROV-O.[9]

The former, once connected to the IoMusT Ontology, defines the `foaf:Agent` as *person, group, software or physical artifact*, and *things that do stuff*. The idea of agent suggested in the previous paragraph is clearly derived from FOAF, although its real utility, in our research, is its capability of including the human being class `foaf:Person` and relationships in the semantic environment. Agents, intended as physical and virtual devices, are described through the latter, PROV-O, where the agent is *something that bears some form of responsibility for an activity taking place, for the existence of an entity, or for another agent's activity* [64]. This idea, in particular, includes also entities running software, which belong to `prov:SoftwareAgent`. Listing 2 shows an example of using FOAF and PROV-O, and introduces in the `iot` namespace the ownership property `iot:owns`.

Listing 2: FOAF & PROV-O integration with the IoMusT Ontology. Extended prefixes are available in Table Table A.3.

```
ns:cristina     a    foaf:Agent, foaf:Person,
                     prov:Agent, prov:Person;
                foaf:name 'Cristina';
                iot:owns  ns:SmartGuitar.
```

Ownership and actual usage do not necessarily coincide: it may happen, for instance, that people use a tool belonging to someone else. Besides, ownership does not imply any sort of activity with the device. A setup for activities, part of the IoMusT Ontology, is available in Fig. 2.

As it can be seen, Fig. 2 contains a rather complex subgraph. First of all, the application introduces the resource URI `ns:bob` as a music performer by exploiting the Music Ontology. The FOAF ontology then provides the `foaf:knows` relationship with other people semantically represented.

Subsequently, by using the `iot` namespace, we start setting up a semantic network to identify the ongoing process involving things and users. In this case the user "Bob" is the subject for the predicate `iot:isInvolvedIn`, that targets a new resource URI with type `iot:Application`. This application class can be explained as the semantic endpoint tagging together all elements, items and agents involved in an activity. A similar description is given by PROV-O documentation for the `prov:Activity` class. Notice that also the device `ns:SmartGuitar` points to the same instance of `iot:Application` accordingly. In addition to this, in order to create the musical background for the IoMusT Ontology, a subclass of the `iot:Application` is suggested for specific IoMusT usage, as reported in Rule 2.

Rule 2. If an `iot:Application` instance is also connected through `iot:isInvolvedIn` to an instance of a class belonging to the Music Ontology, or to the *iomust* namespace, then it is also an instance of `iomust:MusicalThingApplication`.

The application, indeed, is not only a matter of involving the participation of people and objects in an activity. The goal of the IoMusT Ontology is also to represent the application following its sequence of steps over time. Fig. 2 highlights how this is possible

8 http://xmlns.com/foaf/spec/.
9 http://www.w3.org/TR/prov-o/.

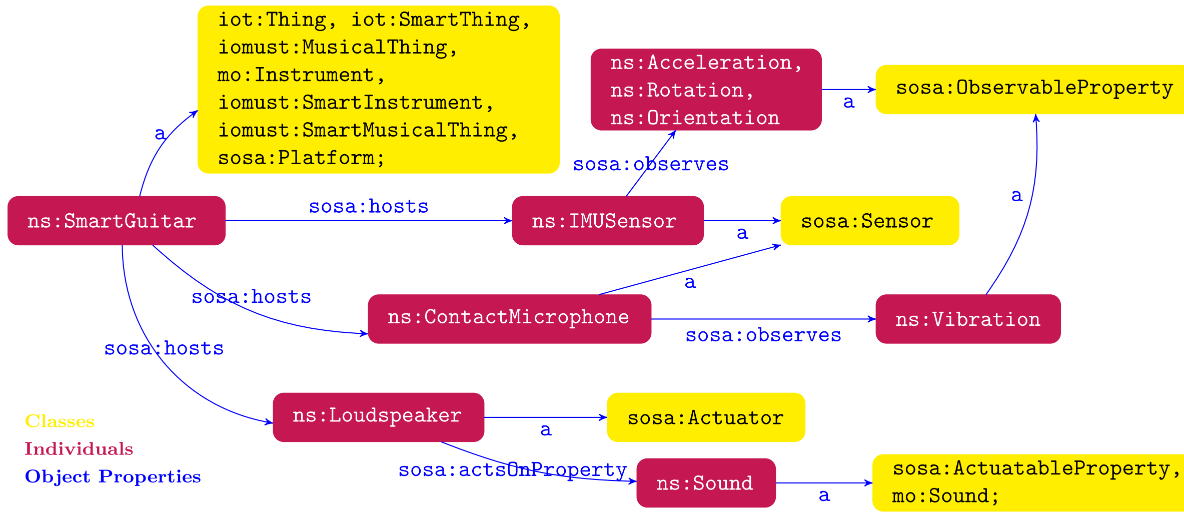

Fig. 1. SOSA integration with the IoMusT Ontology. Extended prefixes are available in Table A.3. The color scheme is the same used in Protégé [58].

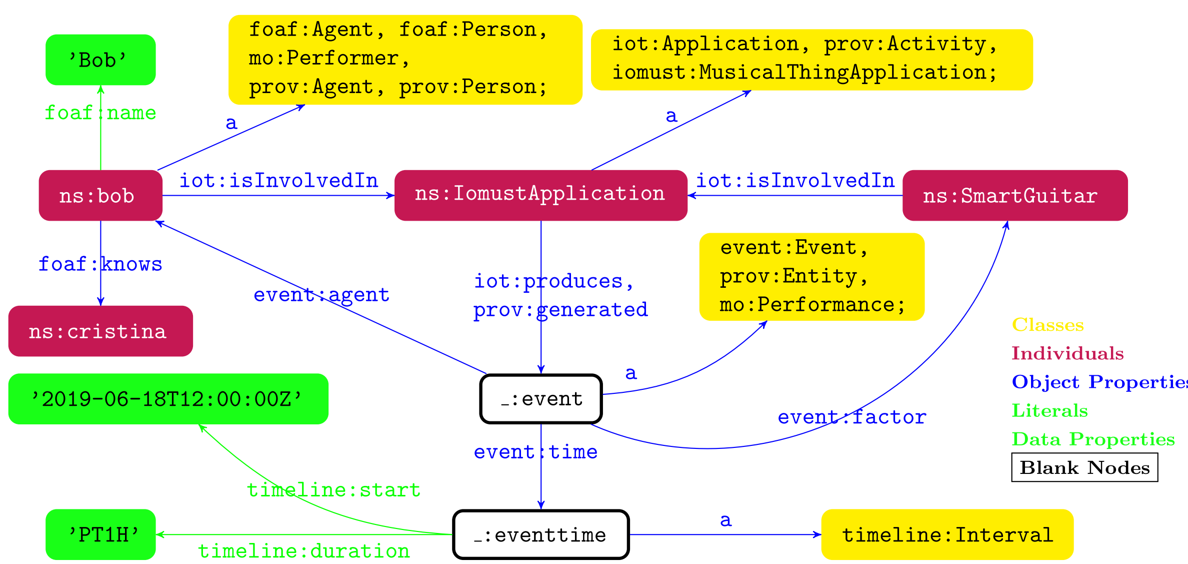

Fig. 2. Activities in the IoMusT Ontology. Undefined resources can be found in previous Figures and Listings. The color scheme is the same used in Protégé [58].

through the usage of the predicate iot:produces. The logic supporting this predicate refers to the application as timed sequence of events, where the *event* is semantically represented by the Event[10] ontology over the event namespace. As it is reported, the event is spawned as a blank node (it may appear on the go), and fully benefits of the predicates available: in a few triples we get full information on the acting agents (e.g., ns:bob), the tools used (e.g., ns:SmartGuitar), and the timings by further addition of the Timeline[11] ontology. Moreover, being the event a source of information, we declare it also as a prov:Entity, alongside with any other information that may be interesting for the user (e.g., the event is a mo:Performance). Summarizing, Listing 2 and Fig. 2 together refer that ns:bob performed some music playing ns:cristina's smart guitar in a performance that lasted 1 h.

10 http://purl.org/NET/c4dm/event.owl#.

11 http://purl.org/NET/c4dm/timeline.owl#.

6.4. Location of devices

Another relevant problem is location of entities in IoT and IoMusT environments. Such piece of information is extremely useful, for example in making spatial statistics on collected data. In order to provide the ontological tools to locate devices, a few considerations follow.

Currently, PROV-O ontology already has an object property devoted to location, namely prov:atLocation. The triple pattern, in such case, is represented in Listing 3 (Example 1) and, as it can be seen, requires the location to be a semantic resource URI. For the example, a DBpedia resource was chosen. To address also situations in which more precision is required, a data property with range xsd:string has been added to the iot namespace, iot:atLocation, that is used in Example 2.

Clearly, other solutions and different approaches are possible.

Listing 3: Location triples alternatives. Extended prefixes are available in Table Table A.3.

```
ns:SmartPiano   a iot:Thing, iot:SmartThing,
    iomust:MusicalThing, mo:Instrument,
    iomust:SmartInstrument, sosa:Platform,
    iomust:SmartMusicalThing, prov:Entity;

[Example 1]
    prov:atLocation dbpedia:London.

[Example 2]
    iot:atLocation
        "51°30'49.3''N 0°05'59.9''W",
        "GW72+F2, London",
        "Paternoster Row, London, UK".
```

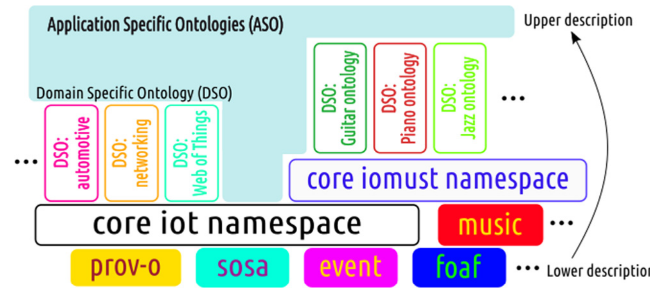

Fig. 3. The IoMusT Ontology is built up incrementally leveraging lower level concepts. It provides the base for other *Domain Specific Ontologies* (DSO) and other Application Specific IoT ontologies (ASO).

7. Implementation and maintenance

The ontology development is accomplished in an online public git repository hosted on GitHub[12] (A.5). The issue tracking system offered by GitHub, will be used as communication channel for maintenance and future development of the ontology (C.3).

The IoMusT vision is structured around several subdomains and related fields, from interfaces for musical expression to the connectivity infrastructure [1]. The creation of an ontology encompassing all the possible facets of the IoMusT domain in all their complexity would be a very significant task that is beyond the scope of this work. For this reason, the IoMusT Ontology is an implementation-driven ontology that is evaluated and evolves during its use while developing applications. This means that the ontology will be growing depending on the appearance of new components around which IoMusT ecosystems are structured, such as novel Musical Things, connectivity infrastructures, or innovative applications and services (F.1). On the technical level, the last version of the ontology will always be accessible at the IoMusT Ontology URI, while past versions will accessible using an URI scheme including the version ID (F.3). For backward compatibility's sake, all the defined concepts will remain in the ontology and keep their current meaning. In case at some point the ontology maintainers decide that a concept is "not to be used any more", it will be annotated as deprecated (F.2).

In its current version, the IoMusT Ontology describes the IoMusT in general terms. As a matter of fact, the work presented in this paper targets a system engineering view enriched with musical content. Consequently, the intent of this research is to provide tools for a global description and easy integration of a new and promising field of IoT. Such premises, as it appears in Section 6, result in a description schema that overviews the IoT in its musical flavor and its higher level features, but does not provide in the examples a taxonomy for the specific devices (i.e., there is no attempt at all to define any form of *Guitar ontology*, *Violin ontology*, and so forth).

Indeed, looking towards the future, it is clear that any musical instrument-specific ontology together with the IoMusT Ontology would represent a set of shared and consistent axioms able to provide a complete semantic approach to internet-connected instruments. Extremely precise discovery over contexts described with a music-professional view may be enabled in this manner.

Looking to Fig. 3, moreover, the forthcoming path is quite easily understandable. First of all, it will be possible to include new lower level vocabularies–taxonomies–ontologies to describe clearly and easily the core `iot` namespace. Secondly, it will be also possible to enrich the `iomust` namespace leveraging both the core `iot` and the new music related ontologies that may appear in the panorama. Eventually, a continuous feedback by developers trying to make innovative and groundbreaking connections between distant fields. *Is the IoMusT Ontology easy to use when it comes to coding? Was it possible to develop your project of connecting the IoMusT Ontology and the new Automotive ontology together?* Implementation and maintenance, in this situation, overlap almost completely.

[12] https://github.com/fr4ncidir/IoMusT.

8. Evaluation

The IoMusT Ontology was assessed by using formal methods as well as checking its fitness for our domain and purposes.

8.1. Metrics and formal validation

Evaluating an ontology is always a matter of identifying the best trade-off between its expressiveness and the performance of applications based on its concepts (i.e., the effective usage). The former is the prevailing aspect in philosophical ontologies, while the latter is of course the most important when dealing with engineering.

Fernández et al. [65] defined twelve metrics to measure the quality of an ontology that we hereby report. In the current paper, not all the metrics have been applied, and some of them required slight modifications to fit the scenario. The reason for this is that, of course, ontology engineering is often a matter of personal interpretation of the designer. Similarly to coding, where evaluation of different implementations and algorithms is made on complexity and performance, the metrics considered relevant for this paper are those belonging to the class of "Knowledge coverage and popularity measures". On the other hand, as IoMusT Ontology is built up as a compound of sub-vocabularies, global metrics are considered less relevant, and will not be included here.

- **Number of classes**: it consists of the number of classes in the analyzed ontology.
- **Number of properties**: this value represent the number of datatype and object properties in a given ontology.
- **Number of individuals**: it is the number of individuals in the ontology.
- **Direct popularity**: this metric represents the number of ontologies importing the given ontology. Being a novel ontology, the popularity is of course equal to zero.
- **Inverse popularity**: the number of well established ontologies, classes and properties imported within the given ontology. It is a way to measure of interoperability with other works vs the novelty introduced, and is calculated on the most basic possible usage (i.e., the one provided in the OWL of the ontology).

Table 2
Evaluation of the IoMusT Ontology according to the "Knowledge coverage and popularity measures" proposed by Fernández et al. [65].

Metric	Value
Number of classes	21
Number of properties	15
- *Datatype properties*	4
- *Object properties*	11
Number of individuals	0
Direct popularity	0
Inverse popularity:	
- *Ontology imports*	7
- *Classes*	29%
- *Properties*	7%

Values for this metric are reported in Table 2.

Based on our previous experience on developing ontologies, metrics belonging to the "structural ontology measures", have been replaced by an alternative set of metrics:

- **Minimum Musical Thing triple count**: the minimum number of triples needed to describe a Musical Thing. According to the previous examples available in Listings 2, 3 and Figs. 1, 2, a very simple Musical Thing can be described with less than 20 triples.
- **Maximum Musical Thing triple count**: this is the maximum number of triples that can be used to describe a Musical Thing. In our case this value is not bounded by the ontology itself, depending on the complexity of the devices although the authors consider that, to the best of their knowledge, it is very unlikely to encounter Musical Things with more than 250 triples.

Classes and properties have been provided with a textual description (`rdfs:comment`) in English (E.7). The ontology editor Protégé [58] and the Visual Notation for OWL Ontologies tool (VOWL) [66] have been used to check the correctness of the ontology. The logical consistency has been checked by running (through Protégé) three reasoners, HermiT (version 1.3.8.413) [67], Pellet (version 2.2.0) [68], and FaCT++ (version 1.6.5) [69] and no inconsistencies have been found.

The evaluation of the ontology went on through the OntOlogy Pitfall Scanner! (OOPS!) online service [70]. This service performs a set of checks to detect common pitfalls in ontology design (based on the existing literature). No major pitfalls have been detected in the IoMusT Ontology. Minor pitfalls have been identified due to: (1) the absence of labels defined through `rdfs:label`; (2) the absence of an inverse relationship; (3) the presence of URIs containing file extensions. As regards the first point, it is ascribable to a design choice: since the ontology (in our opinion) is already easy to read, the adoption of labels would be redundant. The last two points instead, depends on two of the imported ontologies (i.e., the Event and Timeline ontologies).

8.2. Evaluation for requirements and answer to competency questions

Metrics calculation is a good solution to obtain comparable evaluation of ontologies. However, not surprisingly numerical solutions do not take into account the actual topics treated. To address this facet, it is necessary to dive into the ontology, ask questions and evaluate the answers. We hereby suggest three sets of questions, which will be applied to the IoMusT Ontology:

1. The academic community developed over the time some suggestions for ontology engineering. In particular one of the major Conferences for Semantic Web research, namely ISWC, defined in its website[13] a pool of guidelines.
2. MIRO evaluation [21], that provides an organized list of standardized questions. The report[14] of their application to the IoMusT Ontology is available in the ontology's Github repository.
3. Section's 5 competency questions.

Let us start with ISWC guideline analysis, which are also included partially in Miro report. Concerning the *Impact* section, we can definitely say that the IoMusT Ontology fulfills the requests. The answers to the questions were largely discussed over the previous paragraphs of this work, although it is worth repeating that the IoMusT has a dual value, contributing to both the IoT and Music domains. *Reusability*, then, is answered by the explanations given in Section 6, and is maximized by plugging into the IoMusT Ontology well established ontologies like SOSA, FOAF and PROV-O. Eventually, *Design & Technical Quality* and *Availability* are appropriately fulfilled by the concepts provided in Section 7.

Among all evaluations, anyway, the check for competency questions and requirements satisfaction is the most important, because it justifies the whole work. In particular, the 12 competency questions in Section 5.1 are almost completely successfully handled. With the exception of question 4 and 10, the IoMusT Ontology provides all the tools to perform semantic discoveries as complex as needed. So, the ontology provides all the tools necessary to format SPARQL queries that would answer the questions. Question 4, by its side, refers to an aspect that should be treated with the AS ontologies of Fig. 3. Instead competency question 10 may be addressed by a complex discovery including also the concepts of the AudioCommons ontology mentioned in Section 3. However, the AudioCommons ontology has not yet been integrated with the IoMusT ontology, and will be part of a future extension of it.

Concerning Formal Requirements (Section 5.2) the discussion is similar, as some points can be obtained by direct usage of IoMusT ontology as we described it, and some others need the inclusion of additional resources. For example, consider question 5: it is fully achievable by performing SPARQL discoveries as described in the previous paragraph. Competency question 4, on the contrary, refers to live attributes for music, which were not directly targeted here, as they are connected to music and the specific application, and not to devices. Questions 1 and 3 can be achieved by exploiting IoMusT ontology along with specific concepts in the AudioCommons Ontology, Studio Ontology, and Music Ontology. Question 2, then, refers to concepts available in the Studio/Connectivity ontologies.

9. Conclusions

This paper presented in OWL, the IoMusT Ontology, for describing ecosystems forming around IoMusT technologies. The IoMusT Ontology can describe properties of the Musical Things and of the connectivity composing the ecosystem, as well as related applications and services. The creation of the ontology was motivated by the need of facilitating interoperability across heterogeneous Musical Things. The design of the IoMusT Ontology was informed largely by scenarios and use cases present in the IoMusT literature [1,2]. The IoMusT Ontology is related to existing relevant ontologies and models, including the SOSA Ontology [40,41] for the representation of sensors and actuators and the Music Ontology [15] for the authoring and publication of music.

13 http://iswc2018.semanticweb.org/call-for-resources-track-papers/#.

14 https://github.com/fr4ncidir/IoMusT/blob/master/MIRO%20report.md.

Table A.3
Expanded SPARQL prefixes.

Prefix	URI
`rdf:`	http://www.w3.org/1999/02/22-rdf-syntax-ns#
`rdfs:`	http://www.w3.org/2000/01/rdf-schema#
`owl:`	http://www.w3.org/2002/07/owl#
`iot:`	http://www.semanticweb.org/iot/ontologies/2019/5/internet_of_things/
`iomust:`	http://www.semanticweb.org/iot/ontologies/2019/5/internet_of_things/iomust/
`mo, music:`	http://purl.org/ontology/mo/
`prov:`	http://www.w3.org/ns/prov#
`sosa:`	http://www.w3.org/ns/sosa/
`co:`	http://purl.org/co#
`foaf:`	http://xmlns.com/foaf/0.1/
`event:`	http://purl.org/NET/c4dm/event.owl#
`timeline:`	http://purl.org/NET/c4dm/timeline.owl#
`dbpedia:`	http://dbpedia.org/resource/
`ns:`	*whatever personal valid namespace*

Other ontologies were not directly included (e.g., for haptics), but they are still easily integrated in applications whenever needed. This provides easy usage and understanding of the ontology, and at the same time appropriate flexibility.

The evaluation procedure reported in this work showed that the ontology is consistent, follows good practices, and is functional to the ecosystem. Nevertheless, the performed evaluation did not assess the use of the ontology in a real IoMusT setting where heterogeneous Musical Things communicate between each other. In future work we plan to investigate the use of the IoMusT Ontology in an IoMusT ecosystem involving several, distributed, heterogeneous Musical Things connected through a semantic architecture extending those reported in [18,71]. Moreover, we plan to test the ontology with users, based on client applications that make use of it. Furthermore, as the ontology is disseminated more feedback is expected in the near future. These inputs will allow one to evolve the ontology based on potentially unexpected use cases as well as conduct a more in-depth evaluation.

To date, standardization activities for the IoMusT are mostly unrealized [1] and are crucial for its success and indispensable to avoid the fragmentation that characterizes the general IoT field [72]. The work reported in this paper aimed to perform a first step towards this direction.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

G. Fazekas and F. Antoniazzi acknowledge support from the European Union's Horizon 2020 project "Audio Commons", agreement No 688382. The research from F. Giunchiglia has received funding from the European Union's Horizon 2020 FET Proactive project "WeNet – The Internet of us", grant agreement No 823783.

Appendix. Prefixes and namespaces

Table A.3 lists the prefixes used in the present work and their corresponding expanded URI.

References

[1] L. Turchet, C. Fischione, G. Essl, D. Keller, M. Barthet, Internet of musical things: Vision and challenges, IEEE Access 6 (2018) 61994–62017.

[2] L. Turchet, Smart musical instruments: vision, design principles, and future directions, IEEE Access 7 (2019) 8944–8963, http://dx.doi.org/10.1109/ACCESS.2018.2876891.

[3] L. Turchet, A. McPherson, M. Barthet, Real-time hit classification in a smart Cajón, Front. ICT 5 (16) (2018) http://dx.doi.org/10.3389/fict.2018.00016.

[4] L. Turchet, M. Benincaso, C. Fischione, Examples of use cases with smart instruments, in: Proceedings of Audio Mostly Conference, 2017, pp. 47:1–47:5, http://dx.doi.org/10.1145/3123514.3123553.

[5] L. Turchet, M. Barthet, Co-design of musical haptic wearables for electronic music performer's communication, IEEE Trans. Hum.-Mach. Syst. 49 (2) (2019) 183–193, http://dx.doi.org/10.1109/THMS.2018.2885408.

[6] L. Turchet, M. Barthet, Haptification of performer's control gestures in live electronic music performance, in: Proceedings of Audio Mostly Conference, 2019, pp. 244–247.

[7] L. Turchet, T. West, M.M. Wanderley, Smart Mandolin and Musical Haptic Gilet: effects of vibro-tactile stimuli during live music performance, in: Proceedings of Audio Mostly Conference, 2019, pp. 168–175.

[8] M. Wright, A. Freed, A. Momeni, OpenSound control: State of the art 2003, in: Proceedings of the Conference on New Interfaces for Musical Expression, 2003, pp. 153–160.

[9] J. Malloch, S. Sinclair, M. Wanderley, Distributed tools for interactive design of heterogeneous signal networks, Multimedia Tools Appl. 74 (15) (2015) 5683–5707.

[10] J. Malloch, S. Sinclair, M. Wanderley, A network-based framework for collaborative development and performance of digital musical instruments, in: Computer Music Modeling and Retrieval. Sense of Sounds, Springer, Berlin Heidelberg, 2008, pp. 401–425.

[11] J. Malloch, S. Sinclair, M. Wanderley, Libmapper: (a library for connecting things), in: Extended Abstracts on Human Factors in Computing Systems, ACM, 2013, pp. 3087–3090, http://dx.doi.org/10.1145/2468356.2479617.

[12] M. Baalman, H. Smoak, C. Salter, J. Malloch, M. Wanderley, Sharing data in collaborative, interactive performances: the senseworld datanetwork, in: Proceedings of the Conference on New Interfaces for Musical Expression, 2009.

[13] T. Berners-Lee, J. Hendler, O. Lassila, The semantic web, Sci. Am. 284 (5) (2001) 34–43.

[14] J. Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations, vol. 13, Brooks/Cole Pacific Grove, CA, 2000.

[15] Y. Raimond, S. Abdallah, M. Sandler, F. Giasson, The music ontology, in: Proceedings of International Society for Music Information Retrieval Conference, 2007.

[16] G. Fazekas, M. Sandler, The studio ontology framework, in: Proceedings of the International Society for Music Information Retrieval Conference, 2011, pp. 24–28.

[17] A. Allik, G. Fazekas, M. Sandler, An ontology for audio features, in: Proceedings of the International Society for Music Information Retrieval Conference, 2016, pp. 73–79.

[18] L. Turchet, F. Viola, G. Fazekas, M. Barthet, Towards a semantic architecture for internet of musical things applications, in: IEEE Conference of Open Innovations Association, FRUCT, IEEE, 2018, pp. 382–390, http://dx.doi.org/10.23919/FRUCT.2018.8587917.

[19] L. Roffia, P. Azzoni, C. Aguzzi, F. Viola, F. Antoniazzi, T. Salmon Cinotti, Dynamic linked data: A SPARQL event processing architecture, Future Internet 10 (4) (2018) 36.

[20] M. Ulieru, R. Doursat, Emergent engineering: a radical paradigm shift, Int. J. Auton. Adapt. Commun. Syst. 4 (1) (2011) 39.

[21] N. Matentzoglu, J. Malone, C. Mungall, R. Stevens, MIRO: guidelines for minimum information for the reporting of an ontology, J. Biomed. Semant. 9 (1) (2018) 6.

[22] M. Fernández-López, A. Gómez-Pérez, N. Juristo, Methontology: from ontological art towards ontological engineering, in: Proceedings of the Onto- Logical Engineering AAAI-97 Spring Symposium Series, American Association for Artificial Intelligence, 1997.

[23] B. Motik, P. Patel-Schneider, B. Parsia, C. Bock, A. Fokoue, P. Haase, R. Hoekstra, I. Horrocks, A. Ruttenberg, U. Sattler, M. Smith, OWL 2 web ontology language: Structural specification and functional-style syntax, W3C Recomm. 27 (65) (2009) 159.

[24] M. Uschold, Building ontologies: Towards a uni ed methodology, in: Proceedings of 16th Annual Conference of the British Computer Society Specialists Group on Expert Systems, Citeseer, 1996.
[25] A. De Nicola, M. Missikoff, A lightweight methodology for rapid ontology engineering, Commun. ACM 59 (3) (2016) 79–86.
[26] T. Wilmering, G. Fazekas, M. Sandler, The audio effects ontology, in: Proc. of the 14th International Society for Music Information Retrieval Conference, ISMIR'13, November 4-8, Curitiba, Brazil, 2013.
[27] M. Ceriani, G. Fazekas, Audio commons ontology: a data model for an audio content ecosystem, in: International Semantic Web Conference, Springer, 2018, pp. 20–35.
[28] Y. Raimond, F. Giasson, K. Jacobson, G. Fazekas, T. Gangler, S. Reinhardt, The music ontology specification, in: Online Specification Document, 2010, URL http://musicontology.com/.
[29] G. Fazekas, M. Sandler, Knowledge representation issues in audio-related metadata model design, in: Proc. of the 133rd Convention of the Audio Engineering Society, San Francisco, CA, USA, 2012.
[30] B. McBride, Functional Requirements for Bibliographic Records, Final Report, in: UBCIM Publications, New Series, vol. 19, K.G. Saur Verlag, Munich, 1998.
[31] E.T. O'Neill, Frbr: Functional requirements for bibliographic records, Libr. Resour. Techn. Serv. 46 (4) (2011) 150–159.
[32] G. Fazekas, M. Sandler, Describing audio production workflows on the semantic web, in: Proc. of the 14th IEEE International Workshop on Image and Audio Analysis for Multimedia Interactive Services (WIAMIS) 3–5 July, Paris, France, 2013, http://dx.doi.org/10.1109/WIAMIS.2013.6616135.
[33] S. Kolozali, G. Fazekas, M. Barthet, M. Sandler, Knowledge representation issues in musical instrument ontology design, in: Proc. of the 12th International Society for Music Information Retrieval (ISMIR'11) Conference, 24–28 Oct., Miami, Florida, USA, 2011.
[34] S. Rho, S. Song, E. Hwang, M. Kim, COMUS: Ontological and rule-based reasoning for music recommendation system, in: T. Theeramunkong, B. Kijsirikul, N. Cercone, T.-B. Ho (Eds.), Advances in Knowledge Discovery and Data Mining, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-642-01307-2, 2009, pp. 859–866.
[35] E. Myrgioti, V. Chouvardas, A. Miliou, M. Hatalis, Modeling tactile information flow using ontologies, in: 3rd Balkan Conference in Informatics, BCI 07, vol. 2, 2007, pp. 219–228.
[36] E. Myrgioti, N. Bassiliades, A. Miliou, Bridging the HASM: An OWL ontology for modeling the information pathways in haptic interfaces software, Expert Syst. Appl. (ISSN: 0957-4174) 40 (4) (2013) 1358–1371, http://dx.doi.org/10.1016/j.eswa.2012.08.053, URL http://www.sciencedirect.com/science/article/pii/S0957417412010147.
[37] H. Adhami, A. El Saddik, SODHO: Service oriented development of haptics ontology, in: 2014 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE) Proceedings, 2014, pp. 112–117, http://dx.doi.org/10.1109/HAVE.2014.6954341.
[38] B. Albert, J.L. Maire, M. Pillet, C. Zanni-Merk, F.d.B. de Beuvron, C. Knecht, J. Charrier, Generic and structured description of tactile sensory perceptions, in Proceedings of the Kansei Engineering and Emotion Research, Leeds, UK, 2016.
[39] The SSN ontology of the semantic sensor networks incubator group.
[40] K. Janowicz, A. Haller, S.J. Cox, D. Le Phuoc, M. Lefrançois, SOSA: A lightweight ontology for sensors, observations, samples, and actuators, J. Web Semant. (2018).
[41] A. Haller, K. Janowicz, S.J. Cox, M. Lefrançois, K. Taylor, D. Le Phuoc, J. Lieberman, R. García-Castro, R. Atkinson, C. Stadler, The SOSA/SSN ontology: a joint wec and OGC standard specifying the semantics of sensors observations actuation and sampling, in: Semantic Web, vol. 1, IOS Press, 2018, pp. 1–19.
[42] V. Charpenay, S. Käbisch, H. Kosch, Introducing thing descriptions and interactions: An ontology for the web of things, in: SR+ SWIT@ ISWC, 2016, pp. 55–66.
[43] F. Serena, M. Poveda-Villalón, R. García-Castro, Semantic discovery in the web of things, in: International Conference on Web Engineering, Springer, 2017, pp. 19–31.
[44] F. Viola, A. Stolfi, A. Milo, M. Ceriani, M. Barthet, G. Fazekas, Playsound.space: enhancing a live performance tool with semantic recommendations, in: Proc. 1st SAAM Workshop, ACM, 2018, pp. 46–53.
[45] F. Antoniazzi, F. Viola, Building the semantic web of things through a dynamic ontology, IEEE IoT J. 6 (6) (2019) 10560–10579.
[46] L. Turchet, J. Pauwels, C. Fischione, G. Fazekas, Cloud-smart musical instrument interactions: Querying a large music collection with a smart guitar, ACM Trans. Internet of Things (2020) in press.
[47] L. Turchet, M. Barthet, An ubiquitous smart guitar system for collaborative musical practice, J. New Music Res. 48 (4) (2019) 352–365, http://dx.doi.org/10.1080/09298215.2019.1637439.
[48] L. Turchet, A. McPherson, M. Barthet, Co-design of a smart Cajón, J. Audio Eng. Soc. 66 (4) (2018) 220–230, http://dx.doi.org/10.17743/jaes.2018.0007.
[49] K. Siegemund, E.J. Thomas, Y. Zhao, J. Pan, U. Assmann, Towards ontology-driven requirements engineering, in: Workshop Semantic Web Enabled Software Engineering At 10th International Semantic Web Conference, 2011.
[50] V. Braun, V. Clarke, Using thematic analysis in psychology, Qual. Res. Psychol. 3 (2) (2006) 77–101.
[51] M. Grüninger, M.S. Fox, Methodology for the design and evaluation of ontologies, in: Workshop on Basic Ontological Issues in Knowledge Sharing, 1995, pp. 6.1.–6.10.
[52] A. Bröring, S. Schmid, C.-K. Schindhelm, A. Khelil, S. Käbisch, D. Kramer, D. Le Phuoc, J. Mitic, D. Anicic, E. Teniente, Enabling IoT ecosystems through platform interoperability, IEEE Softw. 34 (1) (2017) 54–61.
[53] X. Liu, O. Baiocchi, A comparison of the definitions for smart sensors, smart objects and things in iot, in: 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON, IEEE, 2016, pp. 1–4.
[54] P. Ciccarese, S. Peroni, The collections ontology: creating and handling collections in OWL 2 DL frameworks, Semant. Web 5 (6) (2014) 515–529.
[55] H. Rijgersberg, M. Van Assem, J. Top, Ontology of units of measure and related concepts, Semant. Web 4 (1) (2013) 3–13.
[56] T.W. Narock, A. Szabo, J. Merka, Using semantics to extend the space physics data environment, Comput. Geosci. 35 (4) (2009) 791–797.
[57] X. Wang, X. Zhang, M. Li, A survey on semantic sensor web: Sensor ontology, mapping and query, Int. J. u-e-Serv. Sci. Technol. 8 (10) (2015) 325–342.
[58] M.A. Musen, the ProtégéTeam, The protégé project: A look back and a look forward, AI Matters 1 (4) (2015) 4–12, http://dx.doi.org/10.1145/2757001.2757003,
[59] N.R. Jennings, Agent-oriented software engineering, in: European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Springer, 1999, pp. 1–7.
[60] P. Leitao, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, A.W. Colombo, Smart agents in industrial cyber–physical systems, Proc. IEEE 104 (5) (2016) 1086–1101.
[61] K. Kravari, N. Bassiliades, A survey of agent platforms, J. Artif. Soc. Soc. Simul. 18 (1) (2015) 11.
[62] J. Hendler, Agents and the semantic web, IEEE Intell. Syst. 16 (2) (2001) 30–37.
[63] J. Lin, S. Sedigh, A. Miller, Modeling cyber-physical systems with semantic agents, in: 2010 IEEE 34th Annual Computer Software and Applications Conference Workshops, IEEE, 2010, pp. 13–18.
[64] T. Lebo, S. Sahoo, D. McGuinness, K. Belhajjame, J. Cheney, D. Corsar, D. Garijo, S. Soiland-Reyes, S. Zednik, J. Zhao, Prov-o: The prov ontology, W3C Recomm. 30 (2013).
[65] M. Fernández, C. Overbeeke, M. Sabou, E. Motta, What makes a good ontology? A case-study in fine-grained knowledge reuse, in: A. Gómez-Pérez, Y. Yu, Y. Ding (Eds.), The Semantic Web, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 61–75.
[66] S. Lohmann, S. Negru, F. Haag, T. Ertl, Visualizing ontologies with VOWL, Semant. Web 7 (4) (2016) 399–419, http://dx.doi.org/10.3233/SW-150200.
[67] R. Shearer, B. Motik, I. Horrocks, HermiT: A highly-efficient OWL reasoner, in: Proc. OWLED 2008, vol. 432, 2008, p. 91.
[68] B. Parsia, E. Sirin, Pellet: An owl dl reasoner, in: Third International Semantic Web Conference-Poster, vol. 18, Publishing, 2004, p. 2.
[69] D. Tsarkov, I. Horrocks, FaCT++ description logic reasoner: System description, Autom. Reasoning (2006) 292–297.
[70] M. Poveda-Villalón, A. Gómez-Pérez, M.C. Suárez-Figueroa, Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation, Int. J. Semant. Web Inf. Syst. 10 (2) (2014) 7–34.
[71] F. Viola, L. Turchet, F. Antoniazzi, G. Fazekas, C minor: a semantic publish/-subscribe broker for the internet of musical things, in: IEEE Conference of Open Innovations Association, FRUCT, IEEE, 2018, pp. 405–415, http://dx.doi.org/10.23919/FRUCT.2018.8588087.
[72] E. Borgia, The internet of things vision: Key features, applications and open issues, Comput. Commun. 54 (2014) 1–31.