A Multilevel Modelling Approach for Tourism Flows Detection

Maria Teresa Rossi
Gran Sasso Science Institute
Computer Science
L'Aquila, Italy
mariateresa.rossi@gssi.it

Martina De Sanctis
Gran Sasso Science Institute
Computer Science
L'Aquila, Italy
martina.desanctis@gssi.it

Ludovico Iovino
Gran Sasso Science Institute
Computer Science
L'Aquila, Italy
ludovico.iovino@gssi.it

Adrian Rutle
Western Norway University of Applied Sciences
Bergen, Norway
Adrian.Rutle@hvl.no

***Abstract*—Application development in the Internet of Things (IoT) faces various issues such as lack of separation of concerns and lack of high-level abstraction to address its large scale and heterogeneity. MDE supports the management of this heterogeneity raising the level of abstraction and thanks to its core operations. Multilevel modelling makes it possible to extend MDE techniques to more than two meta-levels permitting model elements to have a dual type-instance dimension, making it particularly suitable for this application domain. People flow monitoring and detection is one of the hot topics in smart cities projects. In this paper, we exploit MDE techniques, through multilevel modelling approaches, to design the infrastructure supporting a solution part of a comprehensive project related to urban informatics. Moreover, even if we target the people flow monitoring and detection scenario, the provided multilevel approach is open and extensible to further IoT scenarios, to specifically manage the evolutionary nature of the IoT.**

I. Introduction

Application development in the Internet of Things (IoT) faces various issues such as lack of separation of concerns and lack of high-level of abstractions to address both the large scale and heterogeneity [1]. Considering the ubiquity and particularity existing in IoT systems, MDE supports the management of this heterogeneity thanks to its core operations, i.e., model transformations, code generation and model management. Standard MDE approaches rely on two meta-levels: one for definition of concepts and one for instances of these concepts [2]. A standard metamodelling approach forces the specification of a domain in a single meta-level; metamodelling facilities are not available at the model level and if those concepts are needed, they must be explicitly modelled at the metamodel level, resulting in unnecessary accidental complexity due to the type-object anti-pattern [2]. Limitation of standard MDE approaches expose the engineered metamodels to the risk of being not flexible enough to deal with the growth of the IoT technology or strongly coupled to specific domains.

Multilevel modelling makes it possible to extend metamodelling to more than two levels permitting model elements to have a dual type-instance dimension [3]. One of the intents of multilevel modelling is language support for expressing types and their instances. When types are to be changed dynamically, the ability to use re-instantiated instances, i.e. treating them again as types, together with other features of multilevel modelling such as deep instantiation, potency, linguistic extensions (see e.g. [2], [4] for details) confirm that IoT-based applications can be widely supported by multilevel modelling. In this respect, IoT architectures and applications have to be flexible enough to allow advancing from lower to higher levels by adding modules (software and hardware) without redesigning the whole model. This is related to the evolutionary nature of IoT and their applications, thanks to emerging technological trends that will keep on shaping the IoT of the future [5]. For instance, models should be easily extensible to add new device types and/or brands, with possibly new communication protocols, as this is a very typical scenario in the IoT context.

People flow monitoring and detection is one of the hot topics in smart cities projects which are heavily based on heterogeneous infrastructures [6] including IoT, mobile development, as well as web and other types of applications. It allows to count people and related applications, such as movement pattern analysis, places with higher transit of people, transit and stay time in point of interests, etc. These types of applications can be faced in different ways, with different types of approaches and technologies. Modelling an application scenario with traditional techniques can present difficulties for the evolutionary nature and complexity induced by various concepts included. This can lead to engineered metamodels with extreme complexity that can cause hard maintenance and poor understandability.

To better cope with these problems, we applied a multilevel modelling approach to design IoT-based applications, also usable in different contexts (e.g. urban context, protected areas) and domains (e.g. tourism domain, mobility). In particular, we exploit MDE techniques [7], through multilevel modelling approaches [4], to design the infrastructure supporting a solution part of a comprehensive project related to urban informatics. To show the usability of our approach, we instantiate two infrastructures for the flow monitoring and detection in two different contexts, namely urban and natural park. However, the people flow monitoring and detection represents only one possible application scenario. The provided multilevel modelling approach is open and extensible to further IoT scenarios, to specifically manage the evolutionary nature of the IoT.

This paper could be seen as an evidence for benefits of using multi level modeling (MLM) in the IoT field. Even if we focus on the modularization aspect, which in turn could be achieved also using other modularization techniques, we will clarify in the paper when we use complex domain-driven requirements which need deep instantiation and the usage of potencies to satisfy these, that are aspects deeply characterizing multi-level modeling.

Structure of the paper. The rest of the paper is organized as follows. In section II we describe two exemplar scenarios about people flow monitoring and we introduce an overview of the hybrid infrastructure we envisage. Some challenges and open issues related to the IoT environment are discussed in section III. In section IV we introduce our multilevel modelling approach for people flow monitoring, which is then applied to two case studies in section IV. The evaluation of the approach against some peculiar requirements of IoT-based infrastructures is presented in section V. Related works are discussed in section VI, while section VII concludes the paper and lists some future directions.

II. People Flow Monitoring and Proposed Hybrid Infrastructure

In this section we describe two different scenarios of flows monitoring, both running in the tourism context. The tourism sector is characterized by a non-uniform distribution, both in space and time [8]. As a consequence, to understand both the economic impacts of tourism and the offer of touristic services in a given area, it is important to analyze tourist movement and behaviour. This would allow the development of appropriate infrastructure and the provision of accurate services, i.e., transport and cultural services. In the last decade, because of the advent of the Internet of Things, the flows detection has been much more accurate thanks to digital tracking technologies (e.g., GPS data, passive positioning of sensors, bluetooth), which allow collecting large amounts of information regarding tourist movements [9]. This type of datasets can be used to gain insights about the overall development of tourist flows both in space and time.

For instance, let's consider the urban context of a smart city where the municipality wants to get insights about the tourist behaviour in order to provide accurate services. It could be interesting for the municipality to know what places are most visited by tourists both to invest in new activities and for managing public mobility. In this context, the flows monitoring can be done in different ways by using different technologies. Behaviour information can be collected through: IoT sensors exploiting, e.g., WiFi or bluetooth signals; mobile phone data supplied from telephone operators; and data provided by third-parties mobile Apps. Then, interesting information can be extracted, combined and exploited.

A similar scenario can be applied in the context of natural parks in which, however, we have to face different geographical conditions where devices must operate. For instance, in this context it is not possible to collect behaviour information using the same technologies of the previous scenario. This is due to the fact that, in natural parks, telephone coverage or internet connection are not always guaranteed. Thus, it might be possible to install sensors that exploit bluetooth signals to collect data, in those cases in which installing WiFi sensors might not be helpful. Moreover, in contexts like this, it could be interesting to also monitor flows of the fauna in the park. Indeed, this would be possible for those animals wearing VHF (very high frequency) collars emitting a pulsed radio signal. These devices allow us to physically locate animals by exploiting receivers and directional antennas.

To collect data from these areas where we may not have internet connection, gateways can be exploited as hubs. For instance, they can be positioned in areas covered by the internet connection and connected with the sensors installed around the park. In this way data can be gathered directly from the gateways. In the natural park context, we cannot rely on mobile phone data, because of the untrustworthiness of the telephone coverage. A similar reasoning can be done for mobile apps, even if it can be possible to collect data offline and send it when the user is in the range of an internet connection. This last consideration arises the distinction between *active* and *passive* collection of data. Mobile apps allow for an active data collection, since they require that the user is logged in the application and provides some information. IoT technologies and telephone operators, instead, enable a passive data collection, since information are collected without a direct involvement of users, e.g., Bluetooth sensors detect users only because they pass by the sensors with a device (e.g., smartphone, car) with the bluetooth turned on and polling.

Considering the multiple application contexts and domains in which flows detection can be suitable, by exploiting the IoT, the solution we envisage in this paper refers to a multilevel modelling approach allowing modelers to easily compose different infrastructure models for different scenarios.

In other words, every modeled infrastructure should be compliant with the high-level one depicted in figure 1. It shows a hardware and software technological infrastructure made of: 1) IoT sensors for detecting radio signals emitted by smartphones, tablets, private and public vehicles (buses) in different points, strategically identified, close to tourist attractions; 2) an open software infrastructure that, starting from the installed sensors, manages different processes. In a first phase ⓐ, it supports the collection of presence data (e.g., of people, vehicles, bicycles, buses) from the installed sensors. In a second phase ⓑ, the collected data are analysed in order to determine different behavioural patterns and to distinguish citizens (daily presence) from tourists (sporadic presence). The third phase ⓒ supported by the infrastructure consists in the extraction of relevant information, such as the most common routes of tourists, the average time spent in each area, the most likely frequency (e.g. daily, weekly, monthly) whit which they visit specific places, congestion situations (e.g. traffic, people), and the temporal and spatial presence distribution on the territory. From this information it is possible to derive the attractiveness of the territory and, consequently, to guide and adapt the tourist offer (e.g. restaurant services, information points, toilets, public transport) in the points of interest.

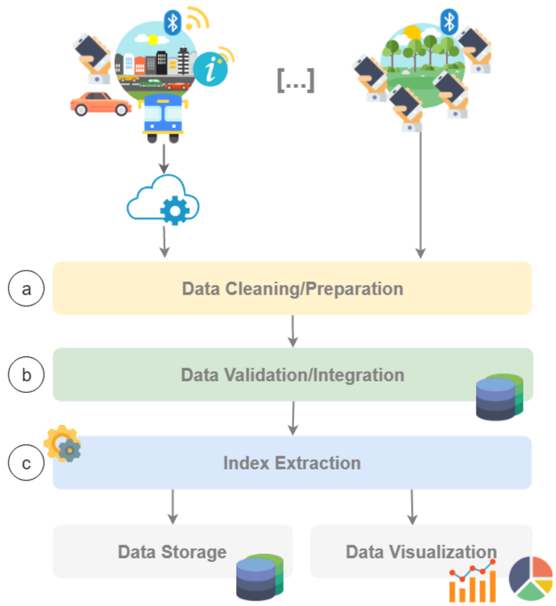

Fig. 1. High-level Overview of the abstract infrastructure

The innovation of this infrastructure mainly lies in the IoT-based data sources. In fact, this helps us to overcome the geographical coverage limitation that might affect telephone operators and mobile applications, since IoT sensors can be installed even in places where this coverage is not guaranteed (e.g., natural parks).

The proposed infrastructure represents a flexible and economical solution that guarantees a constant supply of real time open data. These data can be relevant for different stakeholders in the economic growth of a region, such as public and private entities, transport companies, local governments, and third-parties. Moreover, such infrastructure is meant to be a hybrid one, representing the baseline for the development of applications and services in different areas besides tourism.

III. Challenges and Open Issues

The IoT domain brings new challenges in the modelling of infrastructures and applications relying on the use of devices and hardware components. In this section, we discuss some of these challenges by also arguing about their related benefits and drawbacks.

C1: Heterogeneity of IoT-based data sources. The spread in the use of smartphones led telephone operators and mobile applications become relevant data sources in the development of IoT-based infrastructures and applications. Gaining information from them might be helpful especially when IoT devices are not available, even if they also introduce some disadvantages.

Data provided from *telephone operators* are subject to supply costs and expect a steady dependability from the telephone operators and their end-policies [10]. Although the fact that smartphones are the most used devices nowadays, allowing telephone operators to get a great volume of data, different telephone operators have different market penetrations [10]. This requires estimation over data to be performed. Moreover, the spatial granularity of this data can vary because of the coverage area of the operators antennas [11]. There are also areas which are not covered by any operators (e.g., mountains areas, natural parks, under-developed regions) [10]. Furthermore, telephone operators data are provided in an anonymized way [11], [12] and they are often combined with smartphones usage data, or other data sources, to get composite data [12].

Data from *mobile applications* are collected by third-parties that usually provide them through pay-per-use APIs. Mobile applications can provide a great variety of data types as well as different levels of data accuracy and territorial coverage, depending on the application type [10]. However, differently from telephone operators, the reliability of mobile applications data is not always guaranteed, since it depends on the application providers. Thus, the validation of this kind of data is usually supported by surveys [13]. An alternative solution consists in developing personalized mobile-applications, by supporting the relative effort and costs for the development and maintenance over time. Last but not least, this applications require active users, for contributing to the provided data. Thus, involvement and gamification techniques should be considered in order to gain visibility.

Collecting data from *IoT sensors* requires purchase and installation costs. However, once installed, sensors (and the provided data) belong to the owner and they are no longer linked to the manufacturer (if the architecture allows it). Moreover, the accuracy of these datasets depends on the sensors trustworthiness that, in turn, can depend on the sensors performance and quality level. Exploiting IoT sensors allows data to be collected also from those areas that have not internet/telephone coverage, such as natural parks [14].

Anyhow, to realize IoT-based infrastructures, all these data sources must be taken into account to allow applications to perform as accurate as possible.

C2: Heterogeneity of IoT devices, protocols and platforms. The Internet of Things metaphor emerged rapidly in the last few years, together with the development of new technologies concerning it. Its propagation has enabled a variety of applications that were unbelievable only a few years ago. However, at the same time, several issues also emerged. In particular, we mention a *lack of standardization* and *limited interoperability* [15]. In fact, the IoT is characterized by an increasing number of heterogeneous things, with their private protocols and interfaces. Even those devices offering the same functionality, as for instance presence sensors, belong to different brands providing diverse access modes, data types, accuracy, and so on. Furthermore, the IoT has led to the emergence of several heterogeneous (cloud) platforms, which provide

IoT services enabling the management and interaction with things, such as Amazon AWS-IoT [1], to mention one. Currently, a standard way to interface with things and IoT cloud platform does not exist, and each provider exposes its own APIs. The heterogeneity is rather increasing with the arrival of new things, thus becoming an issue which is far to be solved. Moreover, this would also affect the interoperability among things and IoT services, which is still an open challenge. These aspects require for a significant engineering effort to abstract from the low-level interactions with devices. As a consequence, a further level of abstraction is needed to allow designers to model IoT-based applications and infrastructures.

C3: Low exploitation of MDE approaches in the IoT. The complexity of IoT systems and applications is known. In the near future, things will form large, heterogeneous and highly distributed systems, as envisaged by big market players[2]. In this context, the design and operation of IoT systems and applications become increasingly critical. MDE techniques and approaches might provide a huge support for the development of such systems [7]. Among other things, they would allow designers to manage the before mentioned abstractions, for instance, by exploiting multilevel modelling approaches. Moreover, they would also enable the automation of some of the development activities, such as code generation, model to model transformation and testing. However, the application of MDE in the IoT is still far from being widely practiced [7]. This can be due to the lack of proper requirements that modelling languages must meet to face with the new IoT systems and infrastructure.

IV. Multilevel Modelling for People Flow Monitoring

The proposed multilevel hierarchy of models is shown in figure 2[3]. We engineered our proposal by using the MultEcore tool [16], [17]. Other tools and frameworks could be used for this purpose, however, we decided to use MultEcore at this stage of the development for the following reasons: i) prior experience with the tool and the formalism behind it; ii) the possibility to convert models to Ecore or instances of Ecore models in future for analysis, DSML-creation, code-generation, and other features which come with the EMF out-of-the-box; iii) the need for MultEcore's fine-grained leap potency. Some of the peculiarities of different tools for developing metamodelling hierarchies have been presented in [18]. Here, the authors highlight how MultEcore can deal with some negative aspects of traditional approaches. Indeed, every mentioned tool exploits specific linguistic metamodels with the flattening of the ontological levels. Exploiting MultEcore, instead, does not require custom-made environments and tools and enables the multilevel modelling directly in the Eclipse Modelling Framework (EMF). Multilevel modelling with MultEcore extends the two-level cascading of the other approaches in order to support repeatability. This means that, with this approach, we can repeat the two-level cascading every time we need to create a new model, which is instance of another existing instance. In this way the concept of potency is implemented in a different way w.r.t. the other approaches. In the IoT domain this feature can be widely exploited to satisfy the intrinsic requirements of the domain. Inspired by [19], we define multiple levels of abstractions, from a generic to a specific one, in order i) to do not make the multilevel hierarchy too tied to the target scenario, thus to make it easily extensible, and ii) to incrementally refine general concepts into domain-specific ones. Note that the inter-level relation in our multilevel hierarchy has no strict classification semantics, but the more broad semantics of abstraction or typed-by relation. In the first level, we find the *Communication* component that represents the different communication protocols a *Thing* in the IoT domain can use. The *Communication* also specifies the way in which devices communicate their sensing or receive commands for their actuating through the *sensing* and *actuating* relations, respectively. The potency (red rectangle indicating 1-3-*) for the *Communication* component and its relations specify that we could instantiate Communication up to 3 levels below. At this level, we also find the components related to the concepts of *User* and *Thing*, respectively. In particular, a *Thing* can have two different types of relation with a *User*, namely *detect* and *belongsTo*, to distinguish between specific sensor devices used to detect the user's presence or behavior, from the personal devices belonging to the user. Eventually, given the specific domain we are considering, at this level we further modeled the concept of *Presence*, which can both refer to a user or to a thing (e.g., vehicles). This concept exposes several attributes that describe the detection of a presence (i.e., *user_id, timestamp, lat, long*). The *user_id* can be referred both to people and fauna. Considering the different kind of data and data sources we deal with, as discussed in section III, the abstract concept of *Data* is also designed at this first level of the hierarchy. The *carry* relation between the *Communication* and *Data* concepts represents the fact that data may be exchanged by means of different protocols. Every instance of *Data* has a spatial and temporal granularity. For this reason, we have two attributes, namely *space* and *time*, that embody this granularity in such a way that we can iterate the data in every level of the two dimensions.

In the second level of the hierarchy, we introduce a further type of communication, namely *DataReader*, that performs a specific type of relation, called *access*, with the component *SensedData*. We explicitly define the *DataReader* communication, to distinguish between the protocols used by IoT devices (e.g., bluetooth, WiFi) and a simple read/write access to stored data. Indeed, for *SensedData* we mean the data provided by third-parties (e.g., telephone operators, mobile apps) described by the attribute *origin*, thus already stored in some repository, specified by the attribute *repo_address*. This type of data are often provided in an aggregated way, thus

[1] https://aws.amazon.com/iot/

[2] https://www.ericsson.com/assets/local/news/2016/03/ericsson-mobility-report-nov-2015.pdf

[3] We hide some details to make the picture readable

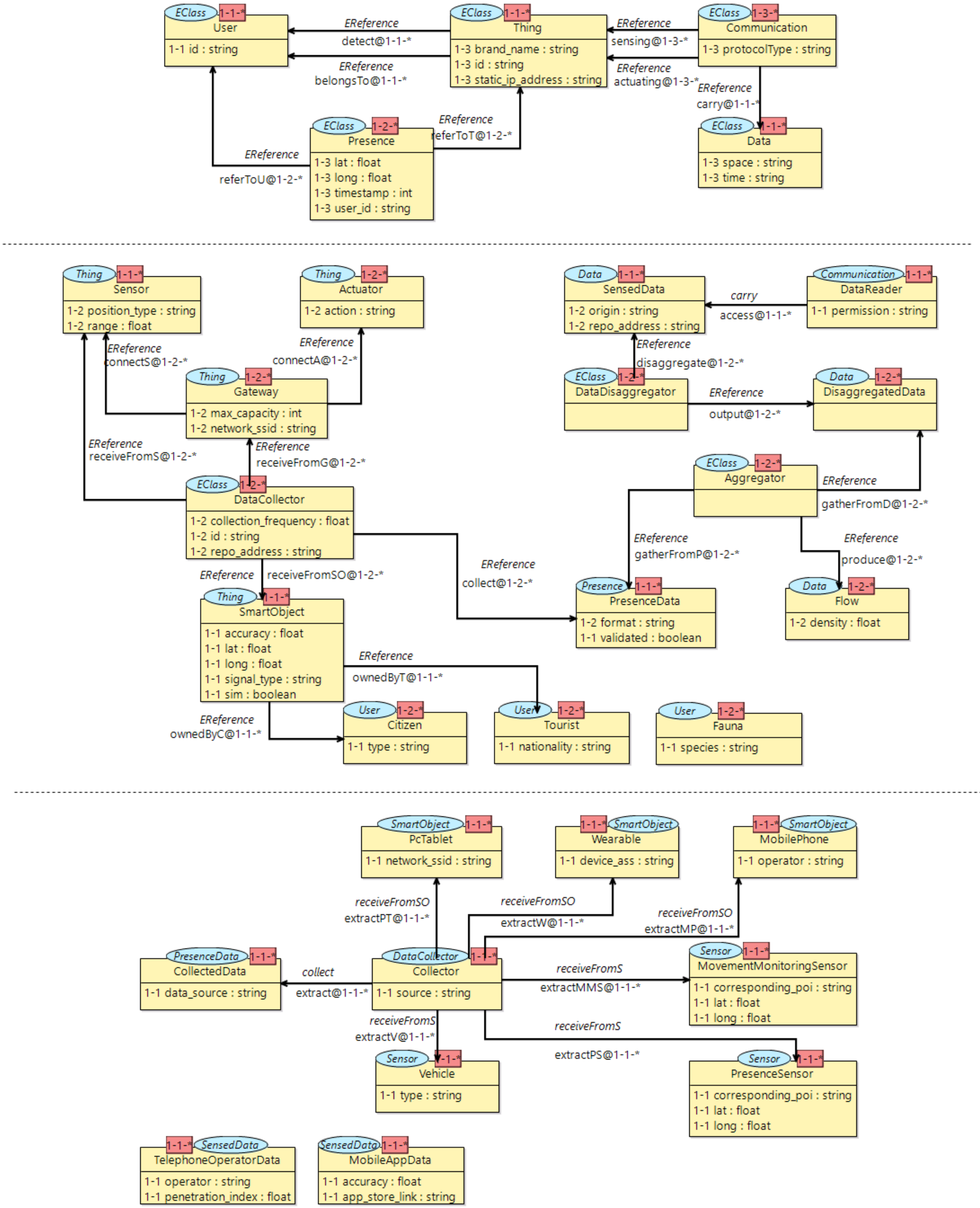

Fig. 2. Overview of the multilevel hierarchical models for supporting Flow Detection.

at this level, we further design a software component called *DataDisaggregator* whose task is to disaggregate this data in such a way that they could be reused as *DisaggregatedData*. Moreover, *PresenceData* can be collected (see *collect* relation) from other data sources (e.g., IoT devices) and, in turn, combined with the *DisaggregatedData* by another software component called *Aggregator*, in order to *produce* value-added information, such as *Flow*. This last concept is an instance of *Data* and, in addition to the inherited attributes, it also contains information about the flow *density*, as a calculated measure. Furthermore, at this level we specialize the concept of *Thing* into the concepts of *Sensor*, *Actuator* and *Gateway* to connect them, if required. *Sensor* has the attribute *position_type*[4] that describes if it is a static or a dynamic sensor and the attribute *range* that describes what is the coverage radius of the sensor. For the *Actuator* we have the attribute *action* that describes the action the actuator can receive. Instead, for the *Gateway* we specify its *max_capacity*, i.e. how many devices can be connected, and the *network_ssid* of the network to which it is connected. Besides these basic devices, in the IoT domain we can also find *SmartObject*s, which can provide smarter services than simply sensing and actuating (e.g., smartphones). Sensors, gateways and smart objects can all provide data to the *DataCollector* software component, with which they might interoperate. This component is specified by *collection_frequency* that describes how often data are collected, in milliseconds, and by *repo_address* that indicates where this data are collected. The *DataCollector*, then, may produce and forward *PresenceData* for the purposes of the *Aggregator*. Also the concept of *User* is refined at this level of the hierarchy. It is specialized in *Citizen*, *Tourist* and *Fauna* in order to distinguish the target of flows detection in our specific scenarios. In particular, we can consider the *type* of users (e.g., resident, commuter), the *nationality* of tourists, in both scenarios, and the *species* of the monitored wild animals, in the natural park context. *Citizen* and *Tourist* can own a *SmartObject*, that is why we have relations between these concepts (i.e., *ownedByC* and *ownedByT*). In this level, we can notice some hardware and software components with potency equals to 2 (e.g., *Gateway* and *Aggregator*) representing those components that can be instantiated at the level of the scenarios models.

In the third level, we instantiate different types of sensors and devices, namely *MovementMonitoringSensor, PresenceSensor, Vehicle, MobilePhone, PCTablet, Wearable*, and different types of *SensedData* (see *MobileAppData, TelephoneOperatorData*). For the concept of sensor we have the indication of the *corresponding_poi*, indicating the point of interest near which the sensor is installed, e.g., an artistic or historical place. Here, the concept of *DataCollector*, previously defined, is instantiated in a *Collector* software component, which contains also the *source* from which it collects data. In this way, the collector component has the information about the origin and destination of the data. In turn, *Collector* is connected with all the devices enabled to provide presence data, through the instantiation of the *receiveFromSO* and *receiveFromS* relations, defined at the second level of the hierarchy between the concepts of *DataCollector*, *SmartObjects* and *Sensor*, respectively. Eventually, the data collected from the different devices can be composed into a *CollectedData*, representing an instance of *PresenceData*. The *PresenceData* has been instantiated here at the third level because it can be extracted only from certain types of sensors (e.g, *PresenceSensor, Vehicle, MobilePhone*). We highlight that, as done for *PresenceData* in the second level of the hierarchy, other types of data might be modeled, e.g., movement data, to represent information coming from other types of sensors (e.g., *MovementMonitoringSensor*). Then, at the third level the *DataCollector* could *collect* further types of data.

In conclusion, we want to highlight here some relevant features of the presented hierarchy and its relation with the peculiarities of IoT domains. First, as we will show in the next section, we applied the multilevel approach in two specific scenarios, taking place in different contexts, namely urban and national park. One could argue why we have not defined two distinct branches, one per each context. This decision relies on the hybrid nature of the proposed infrastructure, which should be exploitable in all the multiple application contexts and domains in which flows monitoring and detection can be suitable by exploiting the IoT. As a consequence, defining separate branches for each context might be a limitation of the approach, even if reasonable.

Moreover, the modeled concepts and relations among them belong to different level of abstractions, without being too tied with the flows detection. In fact, the first level of the hierarchy could be specialized to deal with completely different IoT applications. Indeed, considering that the *Presence* concept is not mandatory (i.e., its relations have a $0..*$ cardinality), the remaining concepts, namely *User*, *Thing*, *Communication* and *Data*, are common in any IoT-based systems and applications.

Exemplary scenario models. In this paragraph we provide two models conforming to the proposed models hierarchy and corresponding to the scenario presented in section II.

Urban scenario. In figure 3 we model an urban context in which we assume to have a presence sensor *SensorX* that captures both *Bluetooth* and *WiFi* signals. From this sensor, we can get two types of *CollectedData*, namely *PeopleData* and *CarData*, through an instance of *Collector*, namely *CollectorA*. In this scenario, we also assume that we can access to some telephone operator data, previously provided by an *OperatorX*, and data gathered by an *infoPoint* mobile application. The two types of sensed data are first disaggregated both by the *DataManipulator* component, and then saved as disaggregated data (i.e., *ThirdPartiesData*). An aggregator component called *IndexExtractor* finally combines the collected and disaggregated data types into output *FlowIndexes*. These indexes will provide information about pedestrian and car flows. The pedestrian flows can be further categorized in those of citizens

[4]In figure 2 we defined as string also those attributes that are of type enumeration. This is due to a current limitation of the MultEcore tool.

and those of tourists.

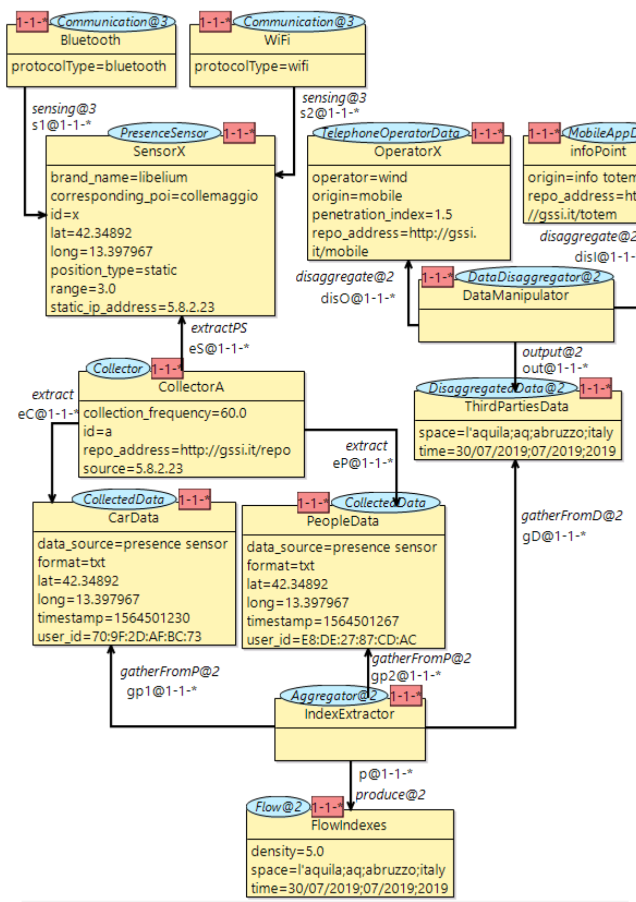

Fig. 3. Model of the urban scenario.

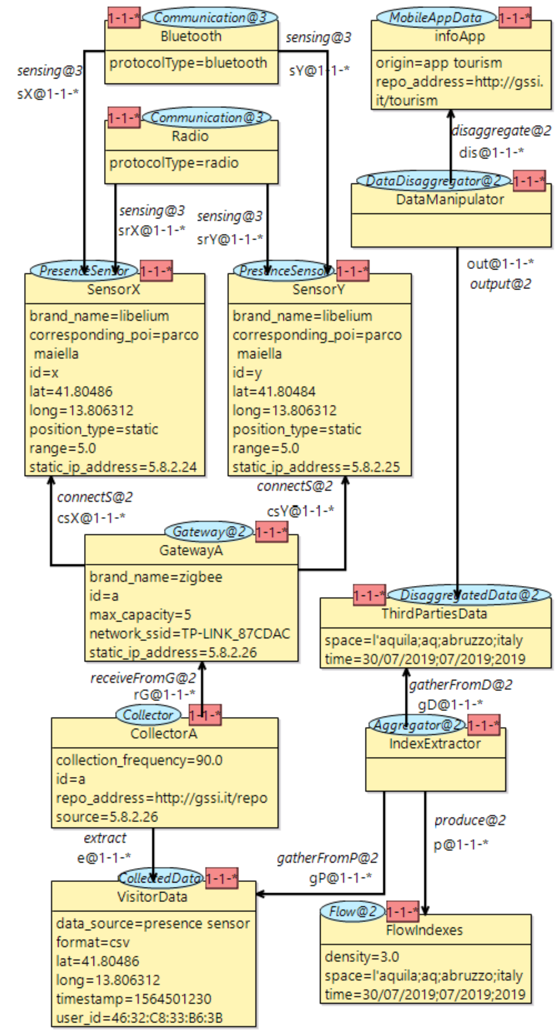

Fig. 4. Model of the natural parks scenario.

Natural parks scenario. In figure 4 we model the scenario of natural parks. Here, to overcome the possible lack of internet connection, we assume to have two presence sensors, *SensorX* and *SensorY*, able to capture *Bluetooth* and *Radio* signals. These sensors are connected to a gateway, namely *GatewayA*, which forwards data from sensors to the *CollectorA* instance. From the installed sensors we extract presence data of park visitors. Here, for visitors we distinguish between citizens and tourists, indifferently, and fauna due to the use of VHF collars tied to animals that emit pulsed radio signals which can be captured by the installed sensors. From the data gathered by *CollectorA* we can obtain a unique dataset, *VisitorData*. Moreover, we also assume to have another data source, which is a mobile app *infoApp* that users might use before or after visiting the natural park, when served by an internet connection. As in the previous scenario, this type of data passes by a *DataManipulator* to be decomposed into *ThirdPartiesData*. Then, the *IndexExtractor* software component produces *FlowIndexes* by making use of all the data provided by the described infrastructure.

We highlight here that for lack of space we designed two simple exemplar scenarios. For this reason, we did not use all the defined concepts in the multilevel hierarchy that, however, can be easily added in the models. For instance, in the urban scenario it would be interesting to monitor *Vehicles*, such as buses, to get public mobility information, besides traffic data. In the natural park scenario, instead, two sensors are not enough to monitor flows, given the wide geographical extent that usually characterizes them. However, we find that the given models are sufficient to show the applicability of the approach in multiple contexts.

In conclusion, in both scenarios made by different settings, we can detect and get indexes about diverse kind of flows (e.g., users, wild animals, vehicles). As shown in figure 1, all the gathered information and related indexes can be stored and visualized in the different applications designed on top of the modeled IoT-based infrastructures.

V. Evaluation

The evaluation of the proposed approach (section V-B), has been performed by analyzing the two scenario models presented in section IV with respect to a set of requirements

IoT-based infrastructures as well as the proposed modelling language have to satisfy (section V-A).

A. Requirements

The requirements we envisage result from the challenges discussed in section III and they are listed in the following:

R1: **Abstraction** of communication protocols, devices and data. An accurate abstraction is needed to deal with the high level of heterogeneity inherent in IoT environments, both as references to data sources (**C1**) and devices with their protocols and platforms (**C2**).

R2: **Flexibility**, with respect to the evolution of IoT technologies. Since the IoT domain is continuously evolving, IoT infrastructures and related design should be able to easily adapt to the changes that could arise [20]. For instance, if a new device brand and/or communication protocol are available (**C2**), perhaps by generating a new data source (**C1**), these changes should be transparent or easy to apply on the infrastructure design, without upsetting its entire structure. This should allow adding a new device belonging to a type already designed in the infrastructure. Moreover a new component for the new device type should be defined and easily added in the infrastructure design, without requiring numerous changes.

R3: **Modularity** of IoT infrastructures. The design of the infrastructure should define loosely coupled components, both inter- and intra-level. This way, it can easily support both the *insensibility* and the *maintainability* of the IoT infrastructure, and hence of its customization. This requirement also supports the feasibility of the previous one. Furthermore, the modularity reduces the cascade of changes when modifications or evolution of the infrastructure must be performed.

R4: **Customizability** and **Reusability** of the modeled infrastructure. From one side, the hybrid infrastructure we envisage should be easily customizable in different contexts and domains, as said in section II. From the other side, the infrastructure should be as generic as possible, thus to enable the reuse of concepts, components and even code among different scenarios. The reusability is, in turn, supported by flexibility and modularity requirements, as for instance in [21].

R5: **Interoperability** between hardware and software components. IoT infrastructures entail, by definition, the involvement of hardware components, besides software ones. This asks for an appropriate support of the communication between these components, in order to guarantee an efficient interoperability among them [20]. Furthermore, often IoT devices do not provide any application logic, which is instead in charge of software components implementing different functionalities on top of the device level. Moreover, it is also interesting to consider the semantic interoperability among heterogeneous physical devices and network, as discussed in [22]. Addressing this requirement with all its facets would help to deal with the high heterogeneity in IoT environment, which causes a limited interoperability [15] (**C2**).

At this point, we guess that the exploitation of MDE approaches, and in particular of multilevel modelling, in IoT (**C3**), might be beneficial for all the listed requirements. In other words, the outlined requirements are generic enough to be easily matched with the peculiarities of MDE approaches (e.g., abstraction). However, at the same time, these requirements are particularly emphasized in IoT systems, due to IoT characteristics. As a consequence, they have a significant impact on modelling languages for managing IoT infrastructures. For instance, the interoperability requirement is further more complex since it has to deal also with hardware components besides software ones, as it was till few years ago. Additionally, the traditional abstraction requirement must now face the wide heterogeneity introduced by the IoT, requiring for more levels of abstraction. It is worth noticing that this first attempt of deriving a modelling language for managing IoT infrastructures points out that we still keep the typical modelling requirements (e.g., interoperability, abstraction), which, at the same time are implicitly made more complex by the IoT features.

B. Scenarios Validation

In this section, we discuss to what extent the multilevel hierarchy of models in figure 2 (i.e., the proposed modelling language) as well as the models we defined for the two considered scenarios, namely urban and natural parks, address the defined requirements.

The abstraction requirement (**R1**) comes with the exploitation of the multilevel modelling approach that enables the possibility of defining more than two levels of abstraction. This feature allowed us to specify, for instance, the concepts of communication protocols, devices and data at a high level of abstraction (see the first level of the hierarchy in figure 2), to refine them level-by-level (e.g., from *Thing*, to *Sensor*, to *PresenceSensor*). The use of the potency construct, introduced in multilevel modelling, further helps in modelling different levels of abstractions, avoiding the implementation of multiple metamodels and supporting generalization.

In both scenarios depicted in figures 3 and 4 it is easy to handle the extension of the designed infrastructure to add new features or devices (**R2**). For example, suppose that we plan to extend our model in order to consider the new Near Field Communication (NFC) technology. NFC devices enable the so-called "touching paradigm" where a service request is triggered by bringing two devices close to each other. To make this evolution, we need to consider the new way devices can communicate, i.e., the touching protocol, in the *protocolType* attribute of the *Communication* component defined in the hierarchy of figure 2. Furthermore, at the third level of the hierarchy we can specify a new component of class *Sensor*, such as *NFC* to model NFC devices. Since these devices can be exploited to collect presence data, we should also define a new instance of the *receiveFromS* relation between the *Collector* and the new *NFC* component. From this example, we can notice how the flexibility of the defined multilevel hierarchy supported us in suitably extending the models without re-defining their entire structure.

Another feature that we can observe in the multilevel hierarchy, which is also reflected in the two scenario models, is the separation between the concepts related to the IoT devices and their data collection, and the sensed data and their manipulation (**R3**). Indeed, we can notice at the second level of the hierarchy how they are organized in two separated sub-modules, connected only by the *collect* relation. This still allow us to implement changes in one of the two sub-modules without affecting the other one. In this way we preserve the maintanability of the IoT infrastructure regardless of its high customizability. In fact, if we make a change in one of the two sub-modules, this would not affect the other one.

The customizability and reusability requirements (**R4**) are easily demonstrated by the two scenario models defined in section IV. Indeed, the urban and natural parks scenarios share the same concepts to model two separate applications customized in two very different contexts. Furthermore, when designing them, we were able to reuse part of the modeled components from one scenario to the other one. For instance, the part of the model devoted to the processing of the data provided by third-parties is pretty the same.

In both scenarios we designed specific software components devoted to both implement the application logic on top of the hardware components (e.g., the *DataCollector* and the *Aggregator* component) and the connection between different devices and networks (e.g., the *Communication* component) (**R5**). This way, we have defined e.g., how to manage data collected by IoT devices or provided by third-parties.

VI. Related Work

In the literature there are some attempts to standardize the development of IoT systems (e.g., [23]–[27]). In these works we can see how MDE can answer the needs in modelling IoT systems. Based on these motivations, we created multiple levels of abstractions by using multilevel metamodels. For example, [23] presents a metamodel-based engineering approach for the systematic development of *SmartObjects* (SO), by proposing four metamodels, each of which corresponds to a different level/phase of the implementation process. Every phase introduces new features and a higher degree of detail in the metamodels. The limitation of this approach is that there is not a concrete connection between the metamodels. Each of them is linked with each other by replicating shared concepts. Because of that, a change in a concept could require the implementation of a series of cascading changes in multiple metamodels. Meanwhile, the multilevel modelling approach allows the inheritance of the changes to the various levels of the metamodel.

In [27] the authors propose a methodology for realizing IoT systems supported by the Model Driven Development (MDD). It is made of four phases with different levels of abstraction, view-point, granularity, and service-orientation. A Smart Environment metamodel framework (SEM) that offers a functional metamodel and a data metamodel is described in [26]. To manage the cooperation between hardware and software, in [24] it is proposed an approach in which the management of sensor devices is abstracted as runtime models. Then, a customized model is constructed according to the specific application scenario and the synchronization between the customized model and sensor device runtime models is ensured through model transformation.

In [25], a Model Driven Architecture (MDA) approach aiming to improve the reusability, flexibility, and maintainability of sensor nodes is proposed. In particular, the authors propose an architecture with different levels of abstractions depending on the development phases of the IoT application (i.e. design, implementation, optimization, and testing).

One of the peculiarities of IoT systems and applications is that of managing infrastructures with both hardware and software components, and highly heterogeneous things. The task of standardization can be very hard. For this purpose, in MDE it has been introduced the concept of family of Domain Specific Languages (DSLs) [28], allowing to deal separately with multiple sub-domains. One of the issues in developing such DSLs is that we can face a wide design space, that could lead us to a low quality of the specification.

What recurs in all these works is that every approach proposes solutions providing multiple unrelated metamodels to implement multiple levels of abstraction of IoT systems. However, this can be overcome by exploiting multilevel techniques.

With regard to work related to multilevel modelling in the literature, there are further attempts to use it as an alternative to traditional modelling approaches. For instance, in [29], a multilevel approach applied to the scenario of user interface development is proposed. Furthermore, in these years, many works have been done to evaluate and prove the efficiency of newly available multilevel modelling approaches (e.g., [30], [31]).

VII. Conclusion

In this work we applied multilevel modelling to a hybrid IoT-based infrastructure for people flow monitoring, usable in different contexts and application domains. Multilevel modelling [4] allowed us to design the proposed infrastructure, by taking advantage of the multilevel modelling paradigm demonstrated with two different case studies. Future works are manifold: firstly, we will explore other application contexts (e.g., smart building) and domains (e.g., mobility) which can benefit by using this approach. Secondly, thanks to the exploration of new IoT contexts and domains, we aim to derive new requirements with which incrementally refine and extend the proposed modelling language. Thirdly, we plan to implement transformations towards other formalisms that can be exploited for specific purposes (e.g., a QoS-based formalism for evaluation purposes) and model-based testing. In the near future, we will use the multilevel hierarchy as prescriptive models for the first applications in an urban informatics project[5].

Acknowledgment

This work was partially supported by the Centre for Urban Informatics and Modelling - National Project - GSSI.

References

[1] P. Patel and D. Cassou, "Enabling high-level application development for the internet of things," *Journal of Systems and Software*, vol. 103, pp. 62 – 84, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121215000187

[2] J. De Lara, E. Guerra, and J. Snchez Cuadrado, "When and how to use multilevel modelling," *ACM Transactions on Software Engineering and Methodology*, vol. 24, pp. 1–46, 2014.

[3] C. Atkinson and T. Kühne, "Reducing accidental complexity in domain models," *Software & Systems Modeling*, vol. 7, no. 3, pp. 345–359, Jul 2008. [Online]. Available: https://doi.org/10.1007/s10270-007-0061-0

[4] C. Atkinson and T. Khne, "The essence of multilevel metamodeling," *UML 2001: łUML 2001 The Unified Modeling Language. Modeling Languages, Concepts, and Tools*, vol. 2185, pp. 19–33, 2001.

[5] L. Atzori, A. Iera, and G. Morabito, "Understanding the internet of things: definition, potentials, and societal role of a fast evolving paradigm," *Ad Hoc Networks*, vol. 56, 12 2016.

[6] M. Alvarez-Campana, G. López, E. Vázquez, V. Villagrá, and J. Berrocal, "Smart cei moncloa: An iot-based platform for people flow and environmental monitoring on a smart university campus," *Sensors*, vol. 17, no. 12, p. 2856, 2017.

[7] F. Ciccozzi and R. Spalazzese, "Mde4iot: Supporting the internet of things with model-driven engineering." *IDC 2016: Intelligent Distributed Computing X*, vol. 678, pp. 67–76, 2016.

[8] F. B. e Silva, M. A. M. Herrera, K. Rosina, R. R. Barranco, S. Freire, and M. Schiavina, "Analysing spatiotemporal patterns of tourism in europe at high-resolution with conventional and big data sources," *Tourism Management*, vol. 68, pp. 101 – 115, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S026151771830044X

[9] N. Shoval and R. Ahas, "The use of tracking technologies in tourism research: the first decade," *Tourism Geographies*, vol. 18, no. 5, pp. 587–606, 2016. [Online]. Available: https://doi.org/10.1080/14616688.2016.1214977

[10] Z. Wang, S. Y. He, and Y. Leung, "Applying mobile phone data to travel behaviour research: A literature review." *Travel Behaviour and Society*, vol. 11, pp. 141–155, 2018.

[11] L. Gabrielli, B. Furletti, F. Giannotti, M. Nanni, and S. Rinzivillo, "Use of mobile phone data to estimate visitors mobility flows." *Software Engineering and Formal Methods*, vol. 8938, pp. 214–226, 2015.

[12] E. Thuillier, L. Moalic, S. Lamrous, and A. Caminada, "Clustering weekly patterns of human mobility through mobile phone data." *IEEE TRANSACTIONS ON MOBILE COMPUTING*, vol. 17, pp. 817 – 830, 2018.

[13] A. Hardy, J. Aryal, and M. Wells, "Comparing techniques for tracking: the case of tourism tracer in tasmania, australia." *E-review of Tourism Research*, vol. 16, pp. 84–94, 2019.

[14] R. Dionisio, P. Marques, S. F.P., J. Rodrigues, and C. Carvalho, "Real-time tourists counting and tracking system for management of nature trails in naturtejo global geopark (portugal)." *Romanian Review Precision Mechanics, Optics and Mechatronics.*, vol. 10, pp. 1–15, 2016.

[15] S. A. Al-Qaseemi, H. A. Almulhim, M. F. Almulhim, and S. R. Chaudhry, "Iot architecture challenges and issues: Lack of standardization," in *2016 Future Technologies Conference (FTC)*, Dec 2016, pp. 731–738.

[16] F. Macías, A. Rutle, and V. Stolz, "Multecore: Combining the best of fixed-level and multilevel metamodelling," in *Proceedings of the 3rd International Workshop on Multi-Level Modelling co-located with ACM/IEEE 19th International Conference on Model Driven Engineering Languages & Systems (MoDELS 2016), Saint-Malo, France, October 4, 2016.*, 2016, pp. 66–75.

[5] The centre for urban informatics and modelling is a Project funded by the Government to the Gran Sasso Science Institute with the intent of supporting the technological transfer in the city of L'Aquila (Italy)

[17] ——, "Multilevel modelling with multecore: A contribution to the MULTI 2017 challenge," in *Proceedings of MODELS 2017, Satellite Event, co-located with ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS 2017), Austin, TX, USA, September, 17, 2017.*, 2017, pp. 269–273.

[18] F. Macías, "Multilevel modelling and domain-specific languages," PhD thesis, Western Norway University of Applied Sciences and University of Oslo, 2019.

[19] S. Sicari, A. Rizzardi, L. Grieco, G. Piro, and A. Coen-Porisini, "A policy enforcement framework for internet of things applications in the smart health," *Smart Health*, vol. 3-4, pp. 39 – 74, 2017.

[20] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani, "Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges," *IEEE Wireless Communications*, vol. 24, no. 3, pp. 10 – 16, 2017.

[21] K. Yelamarthi, M. S. Aman, and A. Abdelgawad, "An application-driven modular iot architecture," *Wireless Communications and Mobile Computing*, vol. 2017, p. 16, 2017.

[22] D. Conzon, P. Brizzi, P. Kasinathan, C. Pastrone, F. Pramudianto, and P. Cultrona, "Industrial application development exploiting iot vision and model driven programming," in *2015 18th International Conference on Intelligence in Next Generation Networks*, Feb 2015, pp. 168–175.

[23] G. Fortino, A. Guerrieri, W. Russo, and C. Savaglio, "Towards a development methodology for smart object-oriented iot systems: a metamodel approach," *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1297–1302, 2015.

[24] X. Chen, A. Li, X. Zeng, W. Guo, and G. Huang, "Runtime model based approach to iot application development," *Frontiers of Computer Science*, vol. 9, p. 540553, 2015.

[25] X. Nguyen, H. Tran, H. Baraki, and K. Geihs, "Frasad: A framework for model-driven iot application development," *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pp. 387–392, 2015.

[26] F. Cicirelli, G. Fortino, A. Guerrieri, G. Spezzano, and A. Vinci, "Metamodeling of smart environments: from design to implementation," *Advanced Engineering Informatics*, vol. 33, pp. 274–284, 2017.

[27] C. Sosa-Reyna, E. Tello-Leal, and D. Lara-Alabazares, "Methodology for the model-driven development of service oriented iot applications," *Journal of Systems Architecture*, vol. 90, pp. 15–22, 2018.

[28] M. Volter and I. Groher, "A family of languages for architecture description," *Proc. 8th Workshop on Domain-Specific Modeling*, 2008.

[29] B. Benner, "A multi-level approach for model-based user interface development," *MODELS-SE 2017*, vol. MULTI - 4th International Workshop on Multi-Level Modelling, pp. 218–225, 2017.

[30] C. Atkinson and T. Khune, "On evaluating multi-level modeling," *MODELS-SE 2017*, vol. MULTI - 4th International Workshop on Multi-Level Modelling, pp. 274–277, 2017.

[31] M. Balaban, I. Khitron, and A. Maraee, "Context-aware factors in rearchitecting two-level models into multilevel models," *MODELS-WS 2018*, vol. MULTI - 5th International Workshop on Multi-Level Modelling, pp. 683–692, 2018.