Modeling Opportunistic IoT Services in Open IoT Ecosystems

Giancarlo Fortino[*], Wilma Russo[*], Claudio Savaglio[*], Mirko Viroli[ǂ], MengChu Zhou[‡]
[*] Dept. of Informatics, Modelling, Electronics and Systems (DIMES), University of Calabria, 87036 Rende (CS), Italy
g.fortino@unical.it, w.russo@unical.it, csavaglio@dimes.unical.it
[ǂ] Dept. of Informatic, Science and Engineering (DISI), Alma Mater Studiorum-University of Bologna, 40126 Bologna, Italy
mirko.viroli@unibo.it
[‡] Dept. of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, 07102, USA
zhou@njit.edu

***Abstract*—Internet of Thing (IoT) is transforming our physical world into a giant information system, daily providing novel, advanced, cyberphysical services. Differently from conventional computing services (e.g., web-services, and ubiquitous services) that are usually loosely impacted by context-awareness, co-location or transience, Internet of Things (IoT) services require to actually consider the overall spatio-temporal context of the heterogeneous entities involved in the service provisioning. This paper proposes a novel and full-fledged approach to IoT service modeling, aiming to fully support IoT service development according to opportunistic properties.**

Keywords—Internet of Things; Cyberphysical Services; Modelling

I. INTRODUCTION

Services notably contributed to the spread of Internet, which evolved from a restricted/small-sized academic and military network into a worldwide platform hosting applications of all kinds [1]. Likewise, services promise to represent the real drivers for the Internet of Things (IoT) [2], a dynamic and heterogeneous ecosystem of networked everyday objects, conventional computing systems, places, pets and people. These entities, supported by ubiquitous and seamless connectivity, take part in novel, advanced, cyberphysical services (indicated as IoT Services in the follow), which are expected to revolutionize every application context. As matter of fact, from industry and public safety, to wellness and transportation, new IoT services are always coming on the scene[1], facilitated by the continuous spread of Smart Objects (SOs, namely everyday objects empowered in their conventional functionalities). Indeed, SOs acquire, process, and communicate information about the surrounding environment, entities and ongoing activities, and accordingly act and interoperate, regardless of their different communication protocols or technologies [3]. In such a scenario, IoT services are therefore fundamental, since they are high-level interfaces for straightforwardly accessing heterogeneous SOs, especially in dense, cooperative, open environments, e.g., a Smart City in which SOs belonging to different application contexts cooperate for providing services related to e-health, smart factories, energy and traffic management, etc. [4].

Although IoT is gaining momentum, and regardless the substantial background on computing services, the development of an IoT service is a challenging and not fully mastered task. Traditional computing services are based on a vertical data flow between physical and application layers, and each service is often independent [1]. Conversely, IoT services exploit both data and cyberphysical functionalities provided by a horizontal landscape of heterogeneous entities, sharing the same resources and environment. Due to their complexity, IoT services require a specific development methodology, so to be thoughtfully designed, formally verified, and simulated. Such a full-fledged approach, so long as supported by a preliminary and systematic modeling phase, paves the way toward reliable, fast and effective IoT Service development [5]. However, service modeling is often a neglected or underestimated activity, which is complicating the overall development process and limiting IoT services potentials.

In this work we propose a novel and full-fledged approach to IoT service modeling, aiming to support IoT Service development according to (i) different granularity, from high-level and general purpose metamodels (suitable to the analysis phase) to detailed models, instantiated over specific domains or case studies (suitable to the implementation and verification phases); and (ii) different perspectives, providing both descriptive and operational service models, thus meeting the requirements of several professionals involved in the service development.

The rest of the paper is organized as follows. In Sec. II, related works pertaining service modeling are surveyed, with a particular focus on the most relevant IoT research initiatives. In Sec. III, our modeling approach for Opportunistic IoT Services is presented and its application shown in a concrete case study (public safety during a mass event's evolution) in Sec. IV. Conclusions are drawn and future work outlined in Sec. V.

II. RELATED WORK

Even though there has been much talk about IoT services, the majority of the related results directly or indirectly derive from only a few IoT Service models.

[1] Using Internet of Things to Create Product-Service Hybrids, - Huawei Publications - http://e.huawei.com/us/publications/global/ict_insights/201502251048/Features/201502251624

One of the most important contributions derives from the IoT-A project [6] (see Fig. 1(a)), in which a detailed IoT service model has been provided and then exploited as an architectural building block ("IoT Service Layer") in different IoT platforms [7] like Butler and ICore. The IoT-A service model extends the previous one developed within the SENSEI[2] project, and is totally aligned with the ones of AIOTI[3] and FIWARE[4] initiatives, as well as with the IEEE P2413 "Standard for an Architectural Framework for the IoT"[5]. According to the SSN (Semantic Sensor Networks)[6] ontology, it describes an IoT service as a well-defined and standardized interface enabling interactions with the real world, specifically through its Virtual Entities (VEs, namely physical entities abstractions). Indeed, IoT services allow accessing a VE's status, properties and functionalities (sensing, actuation, computation, storage or networking) by means of its Resources, thereby hiding VE heterogeneity/complexity to IoT developers and users. Associations between IoT services and VEs are established according to both dynamic (e.g., IoT service current status, VE location, and VE resource availability) and static information (for example, IoT Service specifications and quality of service, VE id and dimension). In particular, relevant information for each IoT service is coded in a Service Description Model according to the business-oriented USDL (Unified Service Description Language). This paves the way toward the application of the IoT-A service model within the world of Business Processes (BPs): indeed, by extending the BPMN 2.0 (Business Process Model and Notation), it is possible to treat IoT services as IoT-aware BPs [8].

Similarly to the IoT-A project [6], authors of [9] and [10] propose an SSN-based model in which IoT services are provided according to established associations between Physical Entities (PE, namely every person, place, or object whose spatio-temporal attributes and preferences constitute its Context). Differently from business-oriented service model of [6], however, the IoT service model of [9] and [10] specifically focus on semantic IoT service description, thus extending the OWL-S (Web Ontology Language for Service)[7]. Indeed, each IoT Service (see Fig. 1(b)) is featured by a ServiceProfile describing what a service does (functional and not-functional properties), a ServiceModel eliciting how a service works (processes and related Preconditions, Effects, Inputs, and Outputs), and a ServiceGrounding specifying how a service is concretely implemented (message formats, serialization, transport and addressing, etc.). In particular, with respect to the original OWL-S service model, ServiceProvisionConstraint, ContextPrecondition and ContextEffect classes have been introduced within the Service Profile to explicitly consider context-awareness and cyberphysicality at the modeling phase. Indeed, the ContextPrecondition class specifies the conditions related to the PE Context (namely its spatio-temporal features) that should hold before the service can be provided (Precondition specifies just general functional preconditions). Similarly, the ContextEffect class describes changes to the external world or environment (Effect just describes the change to the service provider entity). Finally, ServiceProvisionConstraint class represents PE physical constraints that are relevant to the service provision.

An IoT-A like, but not SSN-based, service model is reported in [11] and [12], which mainly consists of IoT services and Entities of Interest (EoI). In particular, the latter represent physical objects, featured by their Properties of Interest (PoI, namely desired properties associated with an EoI), to be monitored, controlled, or tracked through Devices. IoT services, instead, are featured by a set of Requirements which consider a specific application context, an EoI, its PoI, and PoI's observation rate and provided reliability (as shown in Fig. 1(c)). IoT service Requirements are specified in a declarative way and can be autonomously processed and matched with the expected levels of dependability.

A completely different approach to service modeling is carried out in [13] and [14]. In particular, these models are specifically conceived for operational purposes more than for descriptive goals. Indeed, both works exploit (extensions of) Petri Nets [15] to model real world entities as Nets, their operations as transitions and their IoT services as a sequence of states, as shown in Fig. 1(d). Such operational modeling allows controlling the correctness of IoT services among dynamic context changes, thus exhaustively and automatically checking their compliance to a given set of specifications.

Finally, the work in [16] focuses on modelling IoT services at the level of cooperating devices, namely, as computational processes working on spatio-temporal sensed data. The work discussed in the present paper addresses architectural aspects of IoT services sharing a common view where such services operate in given spatio-temporal execution contexts.

III. Opportunistic IoT Service Modeling

This work proposes a novel approach to service modeling, conceived to fully support IoT service development. Our approach has two main steps: (i) metamodeling, in which high-level representations are provided, mainly for descriptive purposes, to outline a service overview particularly suitable for the analysis phase; and (ii) operational modeling, in which services are formalized following specific notations to support the further phases of service design, verification and simulation. These two steps (based on the same concepts but presented from two different perspectives) are both centered around innovative cyber-physical IoT services involving heterogeneous entities, generally defined "IoT Entities", within a certain "IoT Environment", to be detailed later, as depicted in Fig. 2. Similarly to models surveyed in Sec. II, we consider IoT services as interfaces for making an IoT Entity's functionality accessible by other IoT Entities. Conversely, our IoT service model is the first that explicitly considers the following opportunistic properties, crucial to capture the real IoT service potentials but largely overlooked in the past:

i. Dynamicity, IoT services can be dynamically, and not a-priori, created/activated;

ii. Context-awareness, any implicit/explicit information about

[2] SENSEI: Integrating the Physical with the Digital World of the Network of the Future, www.sensei-project.eu
[3] AIOTI: Alliance for IoT Innovation, www.aioti.eu
[4] FIWARE: Future Internet ware, https://www.fiware.org/
[5] IEEE P2413, standards.ieee.org/develop/project/2413.html
[6] SSN, http://www.w3.org/2005/Incubator/ssn
[7] OWL-S, https://www.w3.org/Submission/OWL-S/

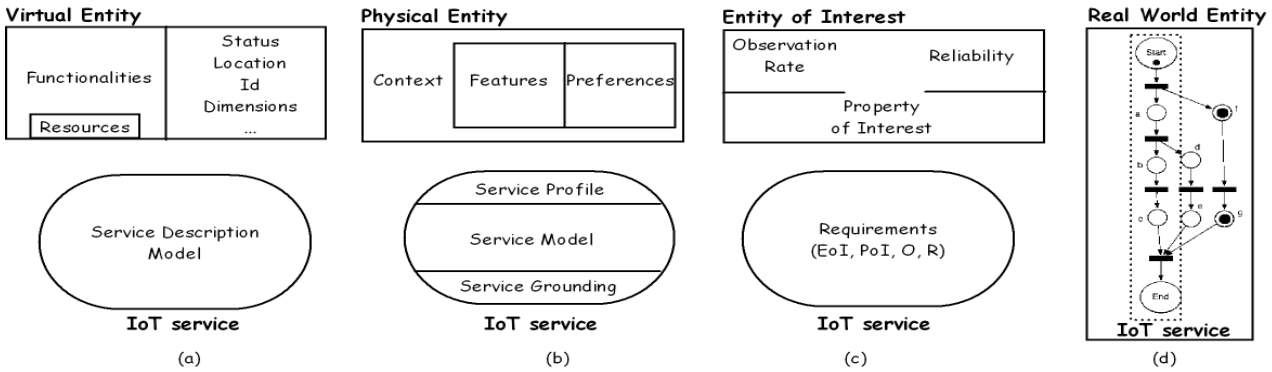


Figure 1 IoT service models in (a) IoT-A [6]; (b) [9-10]; (c) [11-12]; (d) [13-14]

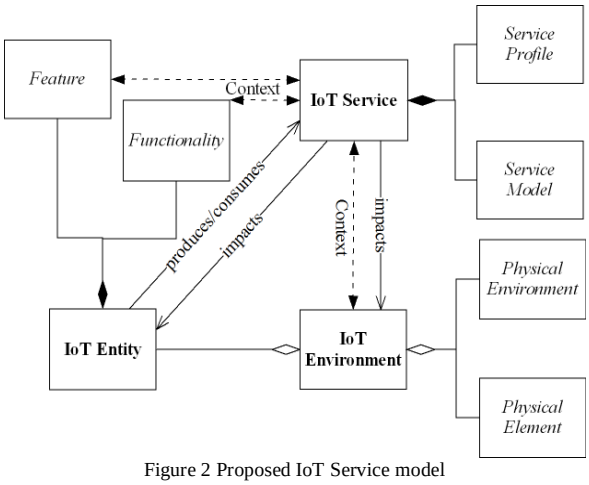


Figure 2 Proposed IoT Service model

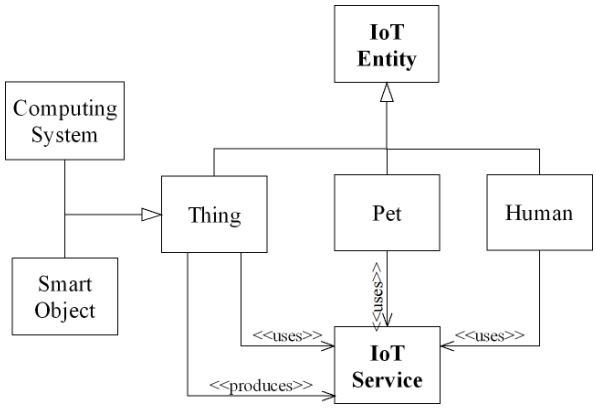


Figure 3 IoT Entities and their roles in IoT service provision

the current location, identity, activity, and physical condition of the involved IoT entities should be considered;

iii. Co-location, IoT services are created for being simultaneously exploited by different IoT entities sharing the same (cyberphysical) resources in the same location;

iv. Transience, IoT services can last for a temporary time or till certain conditions are met.

A. IoT Entity Metamodeling

IoT Entities synergically interact within the IoT Environment, providing and leveraging IoT Services according to their own features (namely static/dynamic attributes) and cyber-physical functionalities (namely entity capabilities subject to specific conditions or constraints). To provide more customized modeling, and differently from the surveyed related works, IoT Entities are categorized into Humans, Pets (both involved uniquely in service consuming) and Things (acting as IoT service prosumers). Fig. 3 depicts the aforementioned IoT Entities' classification and their role in the IoT Service provision. In their turn, Things can be further classified into Smart Objects and Computing Systems. In particular,

Computing Systems are conventional PC, notebooks, servers, etc. They are usually described by means of features like IP/MAC addresses, software and hardware specifications, exposing their functionalities (typically computation) locally or remotely on the Web. Smart Objects (SOs), instead, are everyday objects augmented with sensing/actuation, processing, storing, and networking functionalities. Because of their capabilities, cyberphysical nature and pervasiveness, SOs are primary service prosumers in an IoT scenario.

To consider all the information that could be relevant for the IoT Service provision, the SO metamodel of [17] has been extended in Fig. 4, thus describing each SO through its:

- Status: it is characterized by a list of variables, given as pairs <name, value>, that capture the SO state.
- FingerPrint: it contains the following basic and distinctive SO information, such as Identifier (representing the Id of the SO, which allows its unique identification within the IoT or an IoT subsystem), Creator (either an individual creating the SO for personal use, an industrial company that creates it for business, or an academic research laboratory implementing it for research purposes), Type (given for categorizing SOs with a deeper level of detail, thus distinguishing for example a smart pen from a smart car or a smart building), QoSParameter (associated to the SO, like reliability, availability, etc.), Constraint (defines

an SO static constraint that, if violated, prevent the SO from working, such as electric voltage, and maximum SO work temperature), and Preference (helping choose between alternatives options, properties, modalities, etc. (e.g., a SmartCar with a preferred fuel brand). A preference is not necessarily stable over time and, as opposed to a Constraint, it can be disregarded.

- PhysicalProperty: it represents a physical property of the original object without any hardware augmentation and embedded smartness.
- Service: it models an IoT service provided/consumed by the SO.
- Device: it defines the hardware and software characteristics of a device that allows to augment the physical object and make it smart. A device can be specialized into (i) Computer, representing the SO processing unit (e.g. embedded computer, plug computer, etc.); (ii) Sensor, modeling a sensor node belonging to the SO; and (iii) Actuator: modeling an actuator node belonging to the SO.
- Location: it represents the geophysical position of the SO.

B. IoT Environment and Context Metamodeling

Differently from the conventional computing services, usually loosely impacted by context-awareness, co-location or transience, IoT Services are actually and opportunistically tightly related to the **"IoT Environment"**. It represents the physical environment without any augmentation (e.g., a parking area, an agricultural field, and an industrial warehouse) in which IoT Entities and Physical Elements (e.g., trees, unanimated obstacles, and weather phenomena) are co-located during the IoT Service provision. **Context**, instead, represents a set of dependencies among IoT services and both IoT Entities and the IoT Environment. Indeed, service provision is expected to exploit any implicit or explicit information regarding IoT Entity, IoT Environment, or other IoT Services. For example, an IoT Service can be influenced from an IoT Entity constraint or preference, as well as from the dimensions of the physical environment.

C. IoT Service Metamodeling

Each **IoT Service** is featured by a Service Model and a Service Profile, such that it can be accurately described, automatically discovered, consumed or composed. The Service Model contains the main attributes describing the IoT Service itself and the relationships between the service provision and the involved IoT Environment. In detail:

- Service Name: it refers to the name of the IoT Service that is being offered. It can be used as service's identifier;
- Service Description: it provides a brief human-readable description of the IoT Service;
- Service Category: it refers to an entry in some IoT Service ontology or taxonomy (e.g., monitoring, and payment);
- Service Parameter: it describes the quality parameters provided by the IoT Service (e.g., latency, and precision);
- Service Input: information required for the IoT Service execution;
- Service Output: information generated as output of the IoT Service execution;
- Service Precondition & Service Context Precondition: functional and IoT Entity-related conditions required for a valid IoT Service execution;
- Service Effect & Service Context Effect: events involving IoT Entities which result from the IoT Service execution;
- Service Provision Constraint: IoT Entity's constraint that is relevant to the IoT Service execution.

The Service Profile, instead, contains details about a process, namely the operation(s) concretely implementing the IoT Service. In detail:

- Process Id: it identifies the process;
- Process Input: it specifies the information that the Process requires for its execution;
- Process Output: it specifies the information generated from the Process execution;
- Process Precondition: it specifies the condition under which the Process has place;
- Process Effect: events or changes to the state of IoT Entities that result from the Process execution.

D. IoT Service Operational Modeling

For a number of reasons, IoT services promise to be notably more complicated, heterogeneous and large-scale than conventional ones. First, the IoT service deployment phase is obviously notably complex, time-consuming, and error-prone, comprising not only software distribution but also the configuration of (even thousands of) heterogeneous devices according to their specific resources and surrounding environment [3, 16]. Second, IoT service provisions cannot underestimate several issues related to the network size, density, and topology, as well as failures and changes to service working conditions, that are difficult to be described through static metamodels [5, 16]. Third, IoT services require to completely adhere to their expected provisions, since they perform cyberphysical actuation in time-sensitive, critical environments [9]. It follows that the static and descriptive, yet accurate and expressive, IoT service metamodels need to be complemented by operational IoT service models for paving the way toward verification and simulation phases. DES (Discrete Event System) formalization allows descriptive models to be mapped into operational representations, enabling the subsequent verification and simulation by means of different computing tools. Essential elements in DESs are the (discrete) Event set *Ev* and the (discrete) States Space *Ss*. Ss comprises all the services states (e.g., activation, ready, execution, and aborted) that can be reached according to the possible events (e.g., input received, computed value out of threshold, physical constraint violated, etc.) included in Ev. Doing so, it is possible to model, verify and simulate IoT Services by taking into account relevant elements defining their ServiceProfile and Service Model (e.g., service/process input, output, preconditions, and effects), as well as important IoT Entity features (e.g., constraints, and preferences locations). Petri nets and their extensions (e.g., for dealing with real time and stochastic systems) represent an excellent model for DESs and provide a well-established suite of tools for their formal verification [15], [18]. Future works will also explore advanced

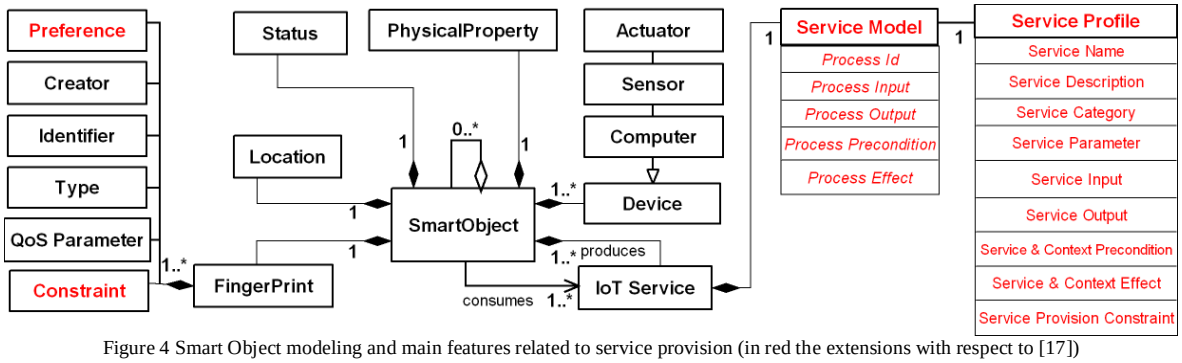


Figure 4 Smart Object modeling and main features related to service provision (in red the extensions with respect to [17])

operational models for large-scale collective adaptive systems, such as the work in [16].

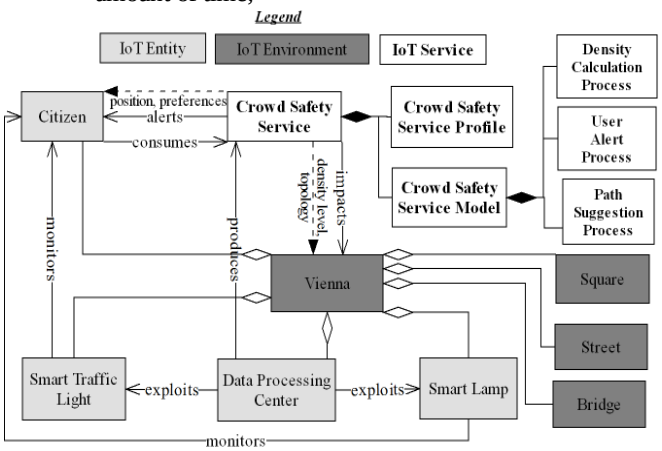
IV. Use Case

The modeling approach described in Sec. III has been applied to the "Crowd safety" opportunistic IoT Service, inspired by the one proposed in [19]. It considers a mass public event, such as the Vienna marathon, and aims at (i) alerting people located nearby overcrowded zones, where any small incident can create a dangerous panic situation; (ii) proposing alternative paths according to the user's preferences/constraints (e.g., a tourist, an elder, a biker can receive different suggestions for the same destination customized on their preferences). In details, SOs deployed around the city (e.g., smart traffic lights, and smart lamps) monitor through their embedded devices the flow of athletes and audience, and allow estimating the city zones' density. The "Crowd Safety" IoT Service" thus alerts citizens located nearby overcrowded zones by sending a notification on their personal devices. The same alerted citizens can hence specify their destination and receive customized, context-aware, and real-time hints on the best path to be followed. The "Crowd Safety" is clearly an opportunistic IoT Service because it exposes the four aforementioned opportunistic properties of:

i. Dynamicity, since it is activated only if a zone's density level exceeds a threshold continuously for a certain amount of time;
ii. Co-located, since it exploits multiple SOs at the same time for contemporary serving multiple citizens located nearby the overcrowded zones;
iii. Transient, since it lasts only for an emergency situation and until the citizen is near an overcrowded zone;
iv. Context-aware, since it considers athletes and audience positions and environmental elements (e.g., a bridge) for determining density and risk levels, as well as citizens positions and their preferences for providing alerts and customized hints.

A. *"Crowd Safety" IoT Service Modeling*

Next, the opportunistic "Crowd Safety" IoT Service is described according to high-level metamodels (Fig.5) and operational models (Fig.6). *Citizens* and *Things* (namely, IoT Entities) located in *Vienna* and deployed on its monitored *Streets*, *Squares* and *Bridges* (IoT Environment) are differently involved in the "*Crowd Safety*" IoT Service. This comprises three processes for mapping each zone to a risk level (*Density calculation*), alerting citizens located near overcrowded zones (*User Alert*), and, if required by the same alerted citizens, providing customized alternative paths for a certain destination (*Path Suggestion*). The Crowd Safety IoT Service and related processes are better detailed through a Service Model and a Service Profile. The former provides functional specifications



Crowd Safety Service Profile

Service Name	**Crowd Safety**
Service Description	Citizens' health safeguard in mass event
Service Category	Public Safety
Service Parameter	Responsiveness = 10 s Accuracy = 50 mt
Service Input	Citizen position, Citizen typology
Service Output	Risk Level
Service Precondition & Context Precondition	Citizen is nearby an overcrowded zone
Service Effect & Context Effect	Citizen is notified on his/her personal device and modifies his/her path
Service Provision Constraint	Target zone is properly monitored by SOs

Density Calculation Process

Process Id	**Density Calculation**
Process Input	Sensed Position
Process Output	Local density value
Process Precondition	Target zone is monitored by SOs
Process Effect	A density value is associated to the monitored zone

User Alert Process

Process Id	**User Alert**
Process Input	User Position, Local density value
Process Output	Risk Level
Process Precondition	User is nearby an overcrowded zone
Process Effect	A notification is sent

Figure 5 Metamodeling of the "Crowd Safety" opportunistic IoT service described according the proposed approach

(e.g., a citizen's position is determined with a precision of 50 meters and they are notified within 10 seconds from their detection near an overcrowded zone), while the second specific preconditions can trigger certain events concretely implementing the "Crowd Safety" IoT Service (e.g., how a city zone gets matched with its density level).

A (simplified but enough expressive) operational model describing the "Crowd Safety" IoT Service according to the Petri net formalism is depicted in Fig. 6. In detail, Service Space *Ss* comprises five service states while six events in Event set *Ev* represent service preconditions (e.g., the density level should exceed a warning threshold for a period before the zone is considered as being overcrowded) and effects (alert notifications or path suggestions are sent to a citizen who is near a dangerous zone). Even at a first glance, it is evident to see the matching between the concepts of Figs. 5 and 6. For example, S_0, S_3 and S_4 depicted in Fig. 6 are the homonyms processes constituting the "Crowd Safety" Service Model in Fig. 5, which encodes, among others, ev_3 as Process Precondition and ev_4 as Process Effect. However, as previously motivated, the metamodels in Fig. 5 accomplish a descriptive functionality while operational model in Fig.6 allows performing the formal verification and simulation of the service.

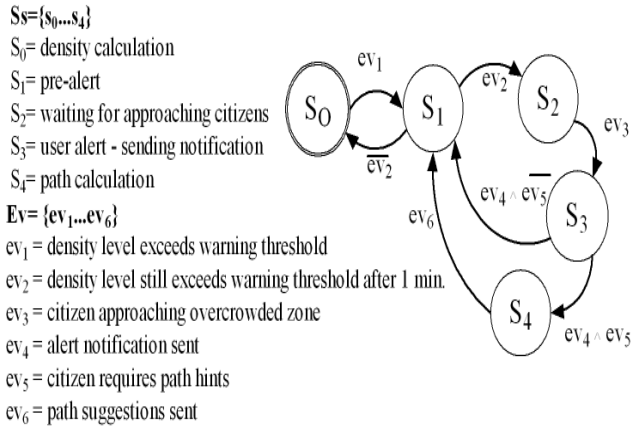


Figure 6 Simplified FSM describing Crowd Safety IoT Service

V. Conclusion

Services are the real IoT drivers, generating unforeseen opportunities into an extremely rich market. The IoT's potential benefits deriving from effectively connected products and services, however, are bounded by some limitations affecting current IoT service development methodologies, especially with regard to IoT service modeling, verification and simulation. In such direction, this work has proposed a novel full-fledged approach that support opportunistic IoT Service development by means of descriptive metamodels and operational models. They are instantiated on a case study related to crowd safety on a large mass event. The approach can be effectively used to analyze, simulate and validate any IoT service before its actual deployment.

Acknowledgment

This work has been carried out under the framework of INTER-IoT, Research and Innovation action - Horizon 2020 European Project, Grant Agreement #687283, financed by the European Union.

References

[1] M. N. Huhns and M. P. Singh, "Service-oriented computing: Key concepts and principles," *IEEE Internet Comput.*, vol. 9, no. 1, pp. 75–81, 2005.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[3] C. Savaglio, G. Fortino, and M. Zhou, "Towards interoperable, cognitive and autonomic IoT systems: An agent-based approach," in *Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on*, 2016, pp. 58–63.

[4] B. Molina, C. E. Palau, G. Fortino, A. Guerrieri, and C. Savaglio, "Empowering smart cities through interoperable Sensor Network Enablers," in *Systems, Man and Cybernetics (SMC), 2014 IEEE Int. Conf. on*, 2014, pp. 7–12.

[5] G. Fortino, W. Russo, and C. Savaglio, "Agent-oriented modeling and simulation of IoT networks," in *Computer Science and Information Systems (FedCSIS), 2016 Federated Conf. on*, 2016, pp. 1449–1452.

[6] S. Meissner, D. D. NEC, and G. M. TID, "Internet of Things Architecture IoT-A Project Deliverable D2. 1–Resource Description Specification."

[7] C. Savaglio and G. Fortino, "Autonomic and Cognitive Architectures for the Internet of Things," in *Int. Conf. on Internet and Distributed Computing Systems*, 2015, pp. 39–47.

[8] K. Sperner, S. Meyer, and C. Magerkurth, "Introducing entity-based concepts to business process modeling," in *Int. Workshop on Business Process Modeling Notation*, 2011, pp. 166–171.

[9] S. De, P. Barnaghi, M. Bauer, and S. Meissner, "Service modelling for the Internet of Things," in *Computer Science and Information Systems (FedCSIS), 2011 Federated Conf. on*, 2011, pp. 949–955.

[10] J. Huang *et al.*, "Extending service model to build an effective service composition framework for cyber-physical systems," in *Service-Oriented Computing and Applications (SOCA), 2009 IEEE Int. Conf. on*, 2009, pp. 1–8.

[11] K. S. Dar, A. Taherkordi, and F. Eliassen, "Enhancing Dependability of Cloud-based IoT Services through Virtualization," in *Internet-of-Things Design and Implementation (IoTDI), 2016 IEEE First Int. Conf. on*, 2016, pp. 106–116.

[12] S. Haller, A. Serbanati, M. Bauer, and F. Carrez, "A domain model for the internet of things," in *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things, IEEE Int. Conf. on and IEEE Cyber, Physical and Social Computing*, 2013, pp. 411–417.

[13] M. A. B. Ahmadon and S. Yamaguchi, "On service personalization analysis for the internet of me based on PN2," in *Consumer Electronics (ICCE), 2016 IEEE Int. Conf. on*, 2016, pp. 413–416.

[14] M. Davoudpour, A. Sadeghian, and H. Rahnama, "'CANthings'(Context Aware Network for the Design of Connected Things) service modeling based on Timed CPN," in *Semantic Computing (ICSC), 2015 IEEE Int. Conf. on*, 2015, pp. 127–130.

[15] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," *Comput. Netw.*, vol. 57, no. 10, pp. 2266–2279, 2013.

[16] J. Beal, D. Pianini, and M. Viroli, "Aggregate programming for the internet of things," *Computer*, vol. 48, no. 9, pp. 22–30, 2015.

[17] G. Fortino, A. Rovella, W. Russo, and C. Savaglio, "Towards Cyberphysical Digital Libraries: Integrating IoT Smart Objects into Digital Libraries," in *Management of Cyber Physical Objects in the Future Internet of Things*, Springer, 2016, pp. 135–156.

[18] Y. Shi, C. Tian, Z. Duan, and M. Zhou, "Model checking Petri nets with MSVL," *Inf. Sci.*, vol. 363, pp. 274–291, 2016.

[19] R. Casadei, D. Pianini, and M. Viroli, "Simulating large-scale aggregate MASs with alchemist and scala," in *Computer Science and Information Systems (FedCSIS), 2016 Federated Conf. on*, 2016, pp. 1495–1504.