IoTUseCase: A New Concept for the Modeling of Business Information Systems in the Age of IoT

Gaetanino Paolone[1], Romolo Paesani[1], Jacopo Camplone[1] and Paolino Di Felice[2][a]

[1]Gruppo SI S.c.a.r.l, 64100 Teramo, Italy

[2]Department of Industrial and Information Engineering and Economics, University of L'Aquila, 67100 L'Aquila, Italy

Keywords: IoT, Software Engineering, UML, Use Cases, Business Modeling.

Abstract: Today, Business Information System (BISs) are the pillars of business operations. The integration of traditional data with data sensed by Internet of Things (IoT) devices can increase remarkably the future of BISs and their importance for the firms. Conceptual modeling is a fundamental stage in the development process of BISs, since it allows to build an abstraction of the reality, which usually is too complex. Browsing into the published literature, it emerges that so far minimal research has been carried out in the area of conceptual modeling of BISs in the age of IoT. The present position paper discusses the modeling of BISs in the age of IoT, as an evolution of the modeling of pre-IoT BISs. The conceptual modeling perspective this paper refers to concerns the Business Modeling of IoT software systems, in the sense meant by the Rational Unified Process (RUP). It is well-known that RUP is a use-case-driven approach, in fact use cases constitute the foundation for the entire software development process. A new UML stereotype, called `IoTUseCase`, is introduced, formalized and its role within the Business Modeling is discussed. Moreover, the study answers two research questions: (RQ1) Are the electronic devices that collect the data of interest part of the Business model or the System model? (RQ2) Is there a Unified Modeling Language (UML) construct that allows modeling the IoT devices?

1 INTRODUCTION

A Business Information System (BIS) is a combination of software, hardware, and telecommunication networks devoted to collect and process data in order to implement the firm's core processes. Humans are also usually considered a component of BISs. BISs have gained immense popularity in business operations over the years. The integration of traditional data with data sensed by Internet of Things (IoT) devices can increase enormously the future of BISs and their importance for the firms which make use of the information technology.

Conceptual modeling is at the center of the development of BISs, since it is a fundamental abstraction of the most relevant aspects in the reality. The story of conceptual modeling, started in 1970s, looks like to be endless, since conceptual modeling efforts are necessary to support new technologies as they emerge (Storey et al., 2023). The more the daily relevance of the information technology increases for the human beings, the more the conceptual modeling emerges as absolutely vital for the digital world. According to the findings in (Storey et al., 2023), so far minimal research has been carried out in this area for the IoT. The same conclusion is drawn in another very recent mapping study about the state of the art of conceptual modeling of IoT systems (Kohan et al., 2023).

The present position paper discusses the conceptual modeling of BISs in the age of IoT, as an evolution of the modeling of pre-IoT BISs. The adopted perspective comes from fifteen years of dealing with the conceptual modeling of BISs (mostly business Web applications for banking). It is our strong believe that the RUP-oriented approach introduced in (Paolone et al., 2010) can be extended to the conceptual modeling of BISs in the age of the IoT. Of course, we are aware that the soundness of such a feeling needs to be verified against a number of real case studies, as it was done by Paolone and his colleagues after the publication at ENASE 2010 of their position paper (Paolone et al., 2010). In the meantime, any sort of feedback from the community is welcome.

The conceptual modeling perspective this paper refers to belongs to the early stage of the development process of IoT software systems: the Business Modeling, in the sense intended by the RUP (Kruchten,

[a] https://orcid.org/0000-0003-3552-0199

Paolone, G., Paesani, R., Camplone, J. and Di Felice, P.
IoTUseCase: A New Concept for the Modeling of Business Information Systems in the Age of IoT.
DOI: 10.5220/0012681400003687
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 19th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2024)*, pages 458-465
ISBN: 978-989-758-696-5; ISSN: 2184-4895

2003)[1]. This point of view offers a high level of abstraction because, in the Business Modeling, business processes are documented in terms of business use cases. This perspective is independent of the complexity of the architecture of the IoT system as well as it is independent of the application domain, while it is mostly focused on assuring a common understanding among all stakeholders of what business processes need to be supported in the organization.

As recalled above, the RUP is a use-case-driven approach. A new UML stereotype, called `IoTUseCase`, is introduced, formalized and its role within the Business Modeling is discussed. Hence, this study addresses two research questions about the changes that the introduction of the IoT technology (a disruptive technology in the today's world) brings to the software development process:

(RQ1) Are the electronic devices that collect the data of interest part of the Business Model or the System model?

(RQ2) Is there a Unified Modeling Language (UML) construct that allows modeling the IoT devices?

By analysing the similarities and differences of BISs in the age of IoT against the pre-IoT BISs, we are able to answer the previous research questions.

The remaining part of the paper is structured as follows. Section 2 recalls the related work, while Section 3 summarizes the Use-Case-centered methodology proposed in (Paolone et al., 2010) for the development of pre-IoT BISs. Hence, Section 4 proposes an extension of such a previous approach to deal with BISs in the age of IoT. The `IoTUseCase` UML stereotype is the core of the proposal because it allows to model the behavior of the IoT devices. Specifically, the Abstract syntax (i.e., the UML metamodel) of the `IoTUseCase` is proposed and discussed. An `IoTUseCase` represents one specific IoT device. The answer to the two research questions mentioned above is given in Section 5. Section 6 ends the paper and provides an overview of the long-term industrial research project.

2 RELATED WORK

It has been widely remarked that IoT systems are complex. To govern such a complexity and get the maximum possible benefit from them, it is essential to manage the IoT systems through the IT technology. From the literature, it arises that there is an increasing engagement of the scientific community in conducting researches regarding the formalization of methods about the development of software for BISs that include IoT devices (Fahmideh et al., 2022).

The investigation of the existing literature highlights that many scholars have approached the development of software systems that include the IoT starting from the architecture of this category of systems (Patel and Cassou, 2015), (Berrouyne et al., 2022), (Kirchhof et al., 2022), (Fahmideh et al., 2022), (Dias et al., 2022), (Saidi et al., 2023). Such a commitment is definitely complex, as it comprises many heterogeneous elements, such as: devices, communication protocols, computing resources (edge, fog, cloud). Such a complexity is the main reason why, despite the rapid growth in IoT research, a general software engineering approach for the development of IoT systems is still missing. This concept is expressed, for example, in (Geller and de Moura Meneses,)−p.81 as follows: "The usage of software engineering principles for the implementation of IoT systems is still being adapted." A similar message may be found in (Fahmideh et al., 2022).

Modeling languages have gained increasing relevance in the IoT domain as a conceptual tool for raising the level of abstraction in system specification, hence improving the productivity of developers and, at the same time, the quality of the IoT applications. (Arslan et al., 2023) is an up-to-date survey on modeling languages that have been designed for IoT software development. In the study, authors analyze 32 different languages with respect to three categories of requirements, i.e., language definition, language features, and tool support.

Modeling languages can be either general-purpose or domain-specific. UML is the most known and widely used general-purpose language. Such a feature limits its suitability for modeling particular domains, for which Domain Specific Languages (DSLs) may be more appropriate. (Kosar et al., 2016; Mernik, 2017; Thanhofer-Pilisch et al., 2017) present a systematic mapping study on DSLs.

The UML profile specialization mechanism by OMG is a more pragmatic approach than the use of DSLs. Based on this, the UML abstract syntax can be specialized to the domain of interest by using the extension mechanisms defined in the Profiles package of UML (OMG, 2017), namely: stereotypes, tagged values, and constraints. One of the most important benefits of using a UML profile instead of a DSL is that the developer is able to use the existing UML tools (e.g., diagram drawing and code generation) in the development process. UML4IoT is an example proposed for the integration of cyber-physical components into an IoT-driven manufacturing industry (Thramboulidis

[1] (Storey et al., 2023) lists Business Modeling as one of the topics belonging to the Conceptual Modeling domain.

and Christoulakis, 2016).

(Prakash and Prakash, 2022) adopts a completely different approch, characterized by a high level of abstraction. The position expressed in such a study is that a problem-driven specification of an IoT application is highly desirable because it would overcome most of the problems that arise when a low-level perspective is adopted. The paper introduces the notion of Information System of Things as an extension of pre-IoT ISs and proposes a set of concepts about what it is called a Conceptual Model of Things (CMoT), necessary for the description of the IoT system to be developed. An instance of the CMoT gives rise to a Conceptual Schema of Things, an artifact about the system to be developed shared among problem/domain experts. Moreover, it sets the basic guidelines of the system to be implemented, and thus, it represents the meeting point between system designers and system implementors.

The perspective we adopted in the present position paper offers a level of abstraction that is comparable to that in (Prakash and Prakash, 2022).

3 A USE-CASE-DRIVEN METHODOLOGY FOR THE DEVELOPMENT OF BISs

The present position paper relies on the work by Paolone and his colleagues described in (Paolone et al., 2010). In that study, they introduced a RUP-driven methodology for drawing the Business and the System Model. Use case modeling and realization are the pillars of their methodology. Their proposal is centered around the four distinct layers of Figure 1. The objective of the first two layers is to get a complete representation of the business reality under investigation; while the objective of the remaining two layers is representing the software system. In detail, the first layer of Figure 1 concerns the Business Use Cases (BUCs) analysis, which are then specialized into Business Use Case Realizations (BUCRs) in the second layer. Subsequently, a trace operation is carried out to define the system Use Cases (UCs) (third layer), which are then specialized into Use Case Realizations (UCRs) (fourth layer). The latter ones can be implemented by a Java class. In addition, Figure 1 shows the relationship between the four layers with the Computation Independent Model (CIM) and the Platform Independent Model (PIM) widely mentioned in the software engineering literature (OMG, 2003).

The journey started in (Paolone et al., 2010) ended in (Paolone et al., 2020). The latter study describes an automatic process to develop enterprise Web applications within the MDA frame of reference. The pillars of the final proposal are, besides the UCs, class and sequence diagrams, which cover, in sequence, the structure and the behavior of the system to be developed, as well as their interactions. The adopted methodological process provides continuity between Business Modeling, System Modeling, Design, and Implementation.

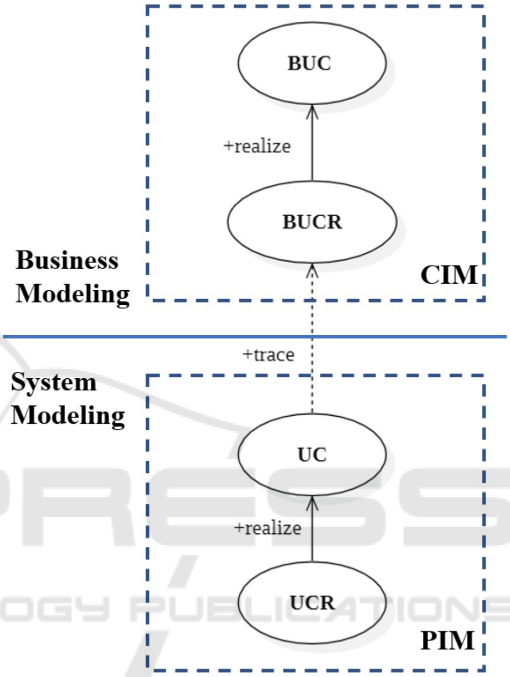

Figure 1: The four methodological layers in (Paolone et al., 2010).

4 DEVELOPMENT OF BISs IN THE AGE OF IoT

The extra challenge in the development of BISs in the age of the IoT, with respect to the pre-IoT case, comes from the need to take into account one or more IoT devices. Our proposal consists in modeling the IoT devices from the point of view of the classes of actors that use them. In detail, we introduce the `IoTUseCase` which is a stereotype of the UML `UseCase`[2] construct, in order to model the behavior of these devices[3]. Each `IoTUseCase` represents one and only one IoT device.

[2] Hereinafter, we join the words Use and Case as in Clause 18 of (OMG, 2017).

[3] "A `UseCase` is a specification of behavior." (OMG, 2017), p.639.

From the methodological point of view, we also propose to bring the IoTUseCase construct, defined within the Business Modeling, till the implementation of the system (hardware and software, as well), in the same way as proposed for the UseCase construct in (Paolone et al., 2010) and (Paolone et al., 2020).

Stereotypes are specific metaclasses that allow designers to extend the vocabulary of UML in order to create new model elements from existing ones but specialized to a domain. The class diagram of Fig.2 shows a simplified version of the Abstract syntax (i.e., the UML metamodel) of the IoTUseCase[4]. The diagram is based on the Abstract syntax of the UseCase construct in Clause 18 of (OMG, 2017), to which the reader is referred to. Below, we describe the IoTUseCase by specializing the description of the UML UseCase construct.

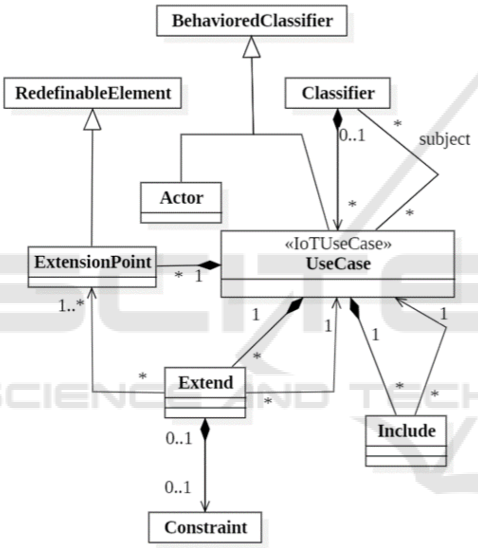

Figure 2: IoTUseCase Abstract Syntax.

The IoTUseCase allows capturing the requirements of an IoT system, i.e., what it is supposed to do. Actors and subjects are two key concepts relatively to IoTUseCases. In this paper, an IoTUseCase's subject represents an IoT device; while Actors represent IoT users (namely humans or other systems) that may interact with a subject.[5] An IoTUseCase is a kind of BehavioredClassifier (i.e., an abstract class) that represents a declaration of a set of offered Behaviors. An IoTUseCase may be owned by a Classifier.

Each IoTUseCase specifies some behavior that an IoT device can perform in collaboration with one or more Actors of a class. IoTUseCases define the offered Behaviors of the IoT device without reference to its internal structure. These Behaviors, involving interactions between the Actors and the IoT device, may result in changes to the state of the IoT device and communications with its environment. An IoTUseCase can include possible variations of its basic behavior, including exceptional behavior and error handling. It is worth mentioning that if UML modelers adopt the level of abstraction we refer to in this paper, then they do not have to take care of several well-known severe issues of IoT devices, namely, limited processor capacity, reduced battery power, interoperability, data and network management, and so on (Dias et al., 2022). This is an objective simplification, with respect to the scenario where the modeling takes place at a lower level of abstraction.

The IoTUseCase can be used both for specification of the market requirements on the IoT device to be selected and for the specification of the functionality offered by the available IoT devices. Moreover, the IoTUseCase may also state the requirements the specified IoT device poses on its environment by defining how the Actors should interact with the IoT device so that it will be able to perform its services. Fig.3 relates the IoTUseCase to the Actors and the two categories of requirements.

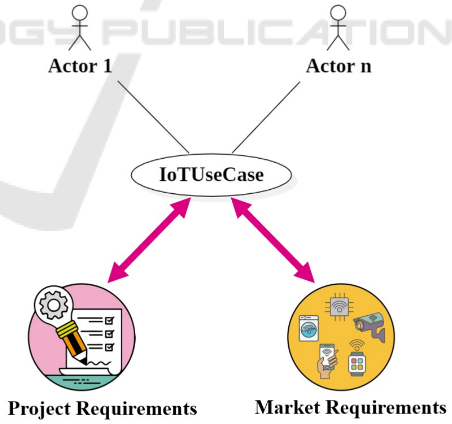

Figure 3: IoTUseCase, Actors and requirements.

Within the context of Business Modeling, we also define the Business IoTUseCase Realization. So, we propose to proceed with the trace operation and, within the System Modeling, we define the System IoTUseCase and the System IoTUseCase Realization. The latter will be implemented in the system to be developed through the chosen IoT device and a software component that allows the use of such a device by a class of actors, according to the mean-

[4] In the diagram, arrows specify, as usual, navigability of the associations. Adornments, dot notations, and association ends are omitted with just one exception: the subject association end because its relevance in this study.

[5] The notion of IoT user is detailed in Sec. 5.

ing of UML Use Case (Booch et al., 1999): “[...] a use case is a description of a set of sequences of actions, including variants, that a system performs to yield an observable result, which is valuable for an actor.”

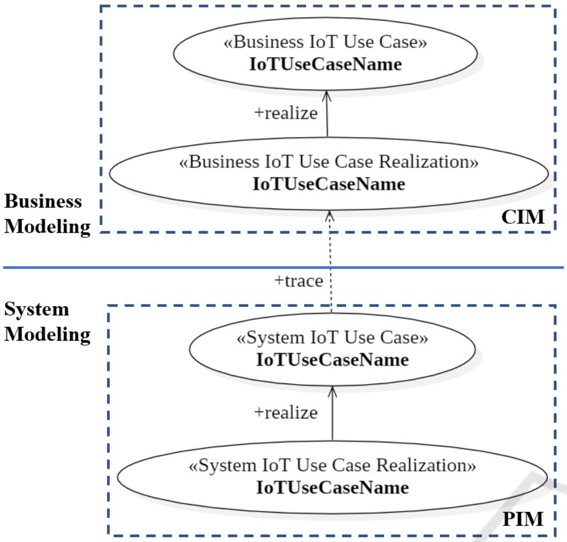

Figure 4: Business & System Modeling of BISs in the age of IoT.

Figure 4 schematizes the mapping between the phases of Business Modeling and System Modeling in the age of IoT as an evolution of pre-IoT BISs.

5 DISCUSSION

Below, the answers to the two research questions listed in Section 1 are given.

(RQ1) Are the electronic devices that collect the data of interest part of the Business Model or the system model?

A fundamental step of Business Modeling consists in eliciting the business use cases. Before writing the use cases, it is helpful to come up with the domain model. The latter forms the foundation of the static part of the model, while the use cases are the foundation of the dynamic part. The static part describes structure; the dynamic part describes behavior.

Domain modeling is the task of building a glossary (i.e., a dictionary of terms) about the main concepts and their relationships specific to the software project to be developed. The purpose of domain modeling is to make sure that all the stakeholders taking part on the software project understand the problem space unambiguously. Referring to a common vocabulary enables clear communication among stakeholders.

Hereafter, we recall briefly the proposal in (Haller et al., 2013) that allows us to provide an answer to RQ2. The IoT Domain model in (Haller et al., 2013) plays an equivalent role of the Conceptual model in (ISO/IEC, 2018). Both models are generic, abstract and simple. They introduce the basic concepts about IoT systems to be known about and describe how the concepts relate to each other logically. Both models are introduced by means of a small number of UML class diagrams where two different types of relationships between classes are used: generalization and association.

The IoT Domain model in (Haller et al., 2013) is abstracted in terms of the following five core concepts (which are also part of the Conceptual model in (ISO/IEC, 2018)):

- Augmented Entity (AE): a real-world object (the Physical Entity – PE) and its digital twin (the Virtual Entity – VE);
- User: either a human or a digital artefact (e.g., a machine, a device, a service);
- Device: hardware to interact with (a sensor) or control (an actuator) physical entities. A Device can be composed of Devices; a Device is a Physical Entity. Sensors and actuators are Devices;
- Resource: software component that gives access to information about, or actuation capabilities on, a Physical Entity;
- Service: exposes Resources through an interface and make them available (to applications and other Services).

(Haller et al., 2013) provides details about these five core concepts and their relationships. Figure 5 shows a simplified version of the UML class diagram of the Domain model.

In light of what just recalled, it follows that devices are a central entity/concept in the development of BISs in the age of IoT. So, it is correct to state that they have to be taken into account since the Business Modeling. This answers to RQ1.

(RQ2) Is there a UML construct that allows modeling the IoT devices?

Designing IoT-based software systems is challenging, but their modeling using UML is a candidate to become a resource to overcome this challenge. Despite many scholars have already been engaged in a debate on the role of UML in the modeling of IoT systems, so far a standard has not been established. (Geller and de Moura Meneses,) provides evidence in this sense by citing 11 articles concerning as many proposals to facilitate the modeling of IoT systems via UML. Each of those studies presented an UML

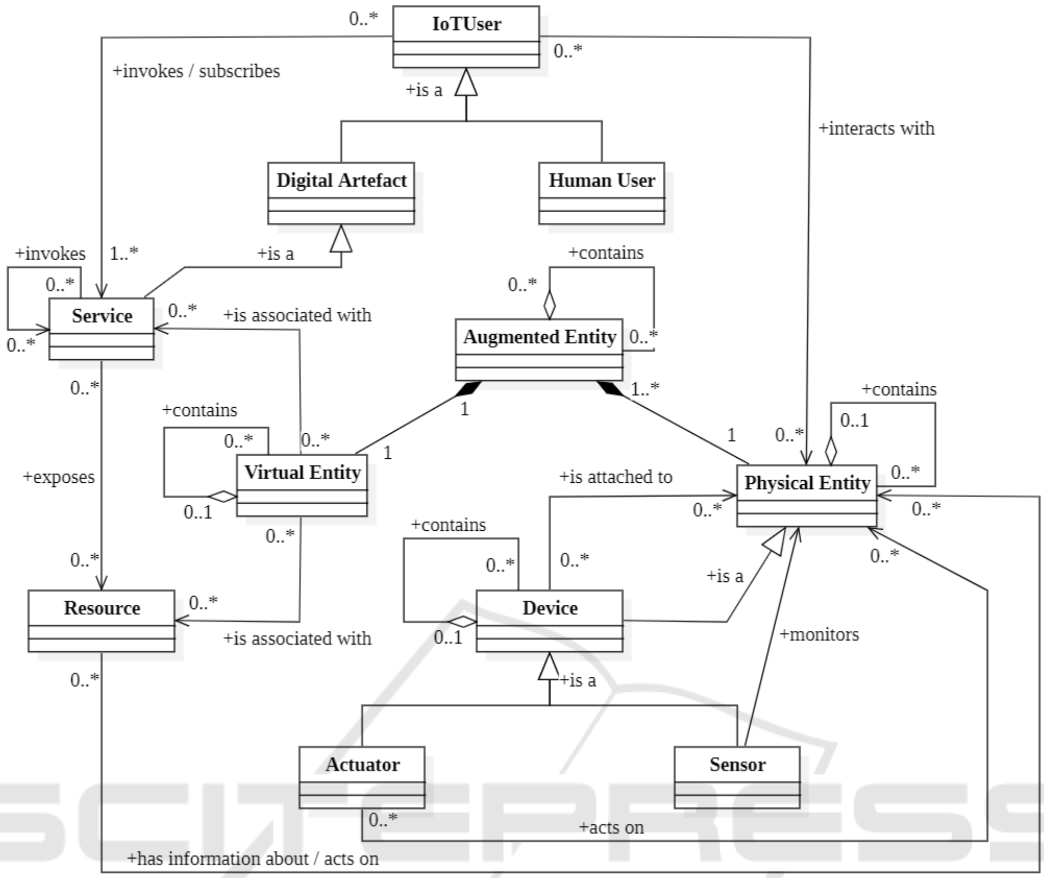

Figure 5: The Domain Model of the IoT.

extension to model IoT systems. Regardless of the solutions that will be available in the coming years, one element emerges loud and clear from the literature, namely that they cannot ignore UML due to the variety of supported models.

In the previous section, it has been argued that to carry out the Business Modeling of IoT systems it is sufficient to introduce the `IoTUseCase` UML stereotype. This answers RQ2.

6 CONCLUSION AND FUTURE WORK

Nowadays the IoT is a notion that denotes a revolution in the information technology that is producing disrupting changes at the information system level. In recent years, information systems developers have been engaged in debate on practical and theoretical issues on the adaptation of information systems to the IoT features. Most scholars approached this domain from the architecture perspective and the infrastructure required for communication. The present position paper introduced, formalized and discussed the notion of `IoTUseCase` (a UML stereotype) at the Business Modeling.

The future work concerns, in sequence, the activities listed below:

- initially, it will be defined a frame of reference regarding the features of the electronic devices commonly used in the development of real-world IoT applications.
- Subsequently, it will be carried out the analysis and the specification (via UML diagrams) of the requirements and software architecture of the component (it will also be called `IoTUseCase`) which implements the abstract `IoTUseCase` described in previous sections.
- Then, the (Java) implementation of the prototype of the `IoTUseCase` software component will be started.
- Finally, the requirements of few case studies devoted to validate the functioning of the aforementioned software component will be formalised. In fact, as part of the experimental activities to be im-

plemented, the industrial partner involved in the research described in this position paper will develop case studies that use the `IoTUseCase` component in specific domains of interest for them. This step is preparatory to the next step in which the development will be aimed at adding intelligence to the aforementioned prototype applications. This extension will be aimed at the development of an AIoT (Artificial Intelligence of Things) application prototype for the management of technological plants.

ACKNOWLEDGEMENTS

This research was funded by B2B S.r.l., 64100 Teramo (Italy).

REFERENCES

Arslan, S., Ozkaya, M., and Kardas, G. (2023). Modeling languages for internet of things (IoT) applications: A comparative analysis study. *Mathematics*, 11, 1263. DOI: https://doi.org/10.3390/math11051263.

Berrouyne, I., Adda, M., Mottu, J.-M., and Tisi, M. (2022). A Model-Driven Methodology to Accelerate Software Engineering in the Internet of Things. *IEEE Internet of Things Journal*, 9(20):19757–19772. DOI: 10.1109/JIOT.2022.3170500.

Booch, G., Rumbaugh, J., and Jacobson, I. (1999). *The Unified Language User Guide*. Addison-Wesley Reading, MA, New York, NY, USA.

Dias, J., Restivo, A., and Ferreira, H. (2022). Designing and Constructing Internet-of-Things Systems: An Overview of the Ecosystem. *Internet of Things*, 19:100529. DOI: https://doi.org/10.1016/j.iot.2022.100529.

Fahmideh, M., Ahmad, A., Behnaz, A., Grundy, J., and Susilo, W. (2022). Software Engineering for Internet of Things: The Practitioners' Perspective. *IEEE Transactions on Software Engineering*, 48(8):2857–2878. DOI: 10.1109/TSE.2021.3070692.

Geller, M. and de Moura Meneses, A. Modelling IoT Systems with UML: A Case Study for Monitoring and Predicting Power Consumption.

Haller, S., Serbanati, A., Bauer, M., and Carrez, F. (2013). A Domain Model for the Internet of Things. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Beijing, China*, pages 411–417. DOI: 10.1109/GreenCom-iThings-CPSCom.2013.87.

ISO/IEC (2018). ISO/IEC 30141:2018, Internet of Things (IoT) - Reference Architecture. https://standards.iso.org/ittf/PubliclyAvailableStandards/c065695_ISO_IEC_30141_2018(E).zip (Accessed on November 25, 2022).

Kirchhof, J., Rumpe, B., Schmalzing, D., and Wortmann, A. (2022). Montithings: Model-Driven Development and Deployment of Reliable IoT Applications. *Journal of Systems and Software*, 183:111087. DOI: https://doi.org/10.1016/j.jss.2021.111087.

Kohan, S., Johnstone, L., and Cetinkaya, D. (2023). Survey on the Model-Centered Approaches to Conceptual Modeling of IoT Systems. In *Frontiers in Computer Science, Section Software*, volume Vol.5. DOI: https://doi.org/10.3389/fcomp.2023.1035225.

Kosar, T., Bohra, S., and Mernik, M. (2016). Domain-Specific Languages: A Systematic Mapping Study. *Information and Software Technology*, 71:77–91. DOI: https://doi.org/10.1016/j.infsof.2015.11.001.

Kruchten, P. (2003). *The Rational Unified Process: An Introduction, 3rd ed.* Addison-Wesley Professional; Addison-Wesley Object Technology Series; Addison-Wesley, New York, NY, USA.

Mernik, M. (2017). Domain-Specific Languages: A Systematic Mapping Study. In Steffen, B. e. a., editor, *SOFSEM 2017: Theory and Practice of Computer Science*, pages 464–472. Springer International Publishing.

OMG (2003). *MDA Guide Version 1.0.1.*

OMG (Dec. 2017). *OMG Unified Modeling Language (OMG UML), Version 2.5.1.* DOI: https/www.omg.org/spec/UML/.

Paolone, G., Di Felice, P., Liguori, G., Cestra, G., and Clementini, E. (2010). A business Use Case Driven Methodology — A Step Forward. In *Proceedings of the 5th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE), 22–24 July, Athens, Greece*.

Paolone, G., Marinelli, M., Paesani, R., and Di Felice, P. (2020). Automatic Code Generation of MVC Web Applications. In *Computers*, volume 9(3), 56. DOI: //doi.org/10.3390/computers9030056.

Patel, P. and Cassou, D. (2015). Enabling high-level application development for the Internet of Things. *Journal of Systems and Software*, 103:62–84. DOI: https://doi.org/10.1016/j.jss.2015.01.027.

Prakash, N. and Prakash, D. (2022). Concepts for Conceptual Modelling of an IoT Application. In *Proceedings of the 17th Inter. Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2022)*, pages 494–501. INSTICC, SciTePress. DOI: 10.5220/0011082800003176.

Saidi, A., Hadj Kacem, M., Tounsi, I., and Hadj Kacem, A. (2023). A Formal Approach to Specify and Verify Internet of Things Architecture. *Internet of Things*, 24:100972. DOI: https://doi.org/10.1016/j.iot.2023.100972.

Storey, V., Lukyanenko, R., and Castellanos, A. (2023). Conceptual Modeling: Topics, Themes, and Technology Trends. In *ACM Computing Surveys*, volume 55 (14s). DOI: https://doi.org/10.1145/3589338.

Thanhofer-Pilisch, J., Lang, A., Vierhauser, M., and Rabiser, R. (2017). A Systematic Mapping Study on DSL Evolution. In *2017 43rd Euromicro Conference on Software Engineering and Ad-*

vanced Applications (SEAA), pages 149–156. DOI: 10.1109/SEAA.2017.25.

Thramboulidis, K. and Christoulakis, F. (2016). UML4IoT—A UML-based Approach to Exploit IoT in Cyber-Physical Manufacturing Systems. *Computers in Industry*, 82:259–272. DOI: https://doi.org/10.1016/j.compind.2016.05.010.