Research Proposal: Collaborative Modeling for IoT

Muzaffar Artikov

Software Engineering Department
Urgench branch of Tashkent University of Information Technologies

April 13, 2018

Abstract. Smart Cities are cities that employ sensors, actors, communication protocols and home servers to increase the efficiency of city services and processes, including water and power supply, waste disposal, police, fire/rescue, hospitals, and schools. Information and Communication Technology (ICT) helps optimizing the processes, while the Internet of Things (IoT) provides the platform for managing a multitude of small sensor and actor devices. A smart city promises not only a more efficient management of resources, but also an increase in the quality of services provided, while still remaining cost-effective.

From a technical point of view, a smart city like any other "Smart X" systems can be seen as a large, sensor-based, distributed information system with data collection, data processing and data support components. Nowadays, smart systems and software architectures are reaching new levels of complexity, necessitating appropriate engineering methodologies. Model-Driven Software Engineering provides the required foundations for formally define generic architectures and development "Smart X" software support on the top. To cope with ever growing complexity of such software architectures, model-driven approaches are extensively applied to model IoT technologies as well as developing and maintaining software and systems platforms for "Smart X" technologies. As the models of these architectures and platforms become huge and complex over time, model-driven collaborative development is necessary to cope with development and evolution challenges arising from the IoT applications. This research proposal aims at (1) studying the state of the art in model-driven IoT and collaborative model-driven IoT, (2) extended research in model-driven IoT focusing on the "Smart X", (3) collaborative model-driven IoT for developing software architectures and platforms for IoT-based systems. As the proof of concepts, it further focuses on applying model-driven collaborative development and maintenance, new upcoming trends in collaborative modeling of IoT, and its adoption in developing smart city architectures.

1 Introduction

Smart cities combine innovative Information and Communication Technologies (ICT) to improve urban services aiming at overcoming social, economic and environmental changes.

The internet of the present day is enriched with a huge amount of aforementioned various devices and micro services. These devices and services operate for different purposes, yet work together to achieve common goals. The various types of sensors, actuators and devices are being developed to provide a range of services. These sensors and devices have created a new technological trend today. This trend has been named the "Internet of Things" [18], "Network of Things" [7] or "Web of Things" [8].

In the framework of IoT, more and more devices are equipped with network connectivity to autonomously provide "smarter" services, forming the Internet of Things (IoT). Applications are wide-ranging, and have variously been termed "Smart X", including Smart Homes, Smart Factories (Industry 4.0), Smart Government, Smart City, Smart Grid, Smart Traffic Control, and many more [11].

Smart Cities are cities that employ sensors and actors to increase the efficiency of city services and processes, including environmental protection/sustainability, energy efficiency, mobility, health care, and safety/security. ICT helps to optimize the processes, while the Internet of Things (IoT) provides the platform for managing a multitude of small sensor, actor devices, communication protocols and home servers.

Internet of Things (IoT) [18] is the basic concept for developing the network of things equipped with built-in technologies for interaction with each other or with the environment. The concept was formulated in 1999 as an understanding of the prospects for the widespread use of radio frequency identification means for the interaction of physical objects with each other and with the external environment.

Definition 1.1 ***Internet of Things***
The Internet of Things is about installing sensors (RFID, IR, GPS, laser scanners, etc.) for everything, and connecting them to the internet through specific protocols for information exchange and communications, in order to achieve intelligent recognition, location, tracking, monitoring and management [reference].

Smart City. The concept of integrating several ICT and IoT solutions for the management of urban property. The purpose of creating a "smart city" is to improve the quality of life with the help of computer technologies to improve the efficiency of services and meet the needs of people. ICT allows the city government to directly interact with communities and urban infrastructure, and monitor what is happening in the city, how the city is developing, and what can improve the quality of life. Through the use of sensors integrated in real time, the accumulated data from urban residents and devices are processed and analyzed.

The concept of smart city arises from the need to manage, automate, optimize and explore all aspects of a city that could be improved and optimized by information technologies. The software paradigm IoT, being a core concept behind smart cities, is largely perceived as a collection of interconnected "things" within smart cities.

The IoT-based smart city applications are realized by interconnected systems of heterogeneous hardware, software, and embedded systems: these cyber-physical systems introduce new levels of complexity, requiring appropriate engineering methodologies to support formally rigorous software and systems development [11]. **Model-Driven Engineering (MDE)** provides fitting foundations and is considered as an enabling technology for advancing Smart X applications.

According to aforemetioned discussions, widely application of ICT to the ordinary cities results in smart cities. A collection of software and hardware components for smart cities can be viewed as a huge reference architecture based on model-driven software platform. Thus, a number of model-driven reference architectures [9] and models [19] are introduced for smart city development, so far. These architectures and models contemplate smart city architectures as a blueprint which provides an appropriate level of abstraction for the development process of smart cities including all aspects of ordinary cities. Model-driven reference architectures and models are used to represent and define different development aspects of smart city architectures gathering different views, viewpoints, software services and components like home servers, communication protocols, sensors, activators, etc., in a single, huge architecture. For instance, several model-driven approaches are investigated in [3] and [17] utilizing different modeling language profiles (e.g. standard UML profiles [14]) in development of smart city architectures.

As a software engineering paradigm, MDE is the modern day style of software and system development which supports well-suited abstraction concepts to development activities. Model-driven engineering intends to improve the productivity of the design and development, maintenance activities, and communication among various actors and stakeholders of a system. In MDE, software models (e.g. in UML) which also comprise source code are the central key artifacts. They are well-suited for designing, developing and producing large-scale software projects. Software models are the documentation and implementation of software systems being developed and evolved. MDE brings several main benefits such as a productivity boost and , models become a single point of truth. Models are reusable and automatically kept in sync with the code they represent [5]. Models are the key artifacts in MDE activities. They are well-suited for designing, developing and producing large-scale software projects. Software models are the documentation and implementation of software systems being developed and evolved [10].

Models are constantly changed during their development and evolutionary life-cycle by various developers and experts. They are constantly evolved and maintained undergoing diverse changes such as extensions, corrections, optimization, adaptations and other improvements. All development and maintenance activities contribute to the evolution of software models resulting in several subsequent revisions. During software evolution, models become large and complex raising a need for collaboration of several developers, designers and stakeholders (i.e. collaborators) on shared models i.e. *Collaborative modeling* [11]. This vision document is a research proposal to work in the field of model-driven IoT and collaborative work IoT-based smart X systems. Its objectives are manifold: (1) studying the state of the art in model-driven IoT and collaborative model-driven IoT, (2) extended research in model-driven IoT focusing on the "Smart X", (3) collaborative model-driven IoT for developing software architectures and platforms for IoT-based systems.

This research proposal is structured as follows: Section 2 motivates the research field defining its contributions and novelty. Section 3 investigates existing model-driven IoT approaches and collaborative modeling approaches for IoT. The main research objectives of this proposal are sketched in Section 4. Section 6 discusses the main expected benefits of the research. This paper ends up in Section 7 by drawing some conclusions.

2 Motivation

A smart city promises not only a more efficient management of resources, but also an increase in the quality of services provided, while still remaining cost-effective. It is very important to keep the citizens as central stakeholders in mind, which is why they must be included in all stages of Smart City development, starting with the planning stage. Smart cities are more flexible than traditional city management approaches in case of unexpected events.

Implementing a smart city requires to understand the complex processes in cities, so that proper conclusions can be drawn from available data. Depending on the specifics, this can be achieved either through human analysis or through artificial intelligence, periodically or in real time. It is also necessary to manage large, distributed and diverse networks of devices. These devices need to communicate, they need a reliable power supply, and they need to be resilient against errors and sabotage. In addition, they produce enormous amounts of real-time data that needs to be analyzed and stored. This requires expertise in many different areas of research and the ability to work on interdisciplinary projects.

All aforementioned challenges of smart distributed systems can be more or less eased by generic reference architectures [5], [7], [9]. From a technical point of view, all IoT-based smart systems can be treated as a large, sensor- and actor-based, distributed information system. These systems can then be generalized as a large, sensor-based software reference architecture [9] which include main activities like data collection by sensors, data processing by servers, control by activators and data support for actors.

MDE provides a collection of modeling concepts and the detailed separation of different views and concerns [9]. For instance, components have well-defined interfaces and ports. MDE benefits from its capabilities to model different, but integrated views and behavior of IoT systems which are eventually made executable for different smart platforms. The basic idea of the smart city vision is the pervasive presence of a variety of things or objects such as sensors, actuators, mobile phones, etc. which are able to interact with each other and cooperate with their pairs to reach common goals [1] [18].

3 State of Art

This section shortly reviews model-driven approaches and methodologies for developing IoT applications.

There are several MDE approaches for developing IoT applications, e.g. the Sirius-based ThingML language [5]. These model-driven approaches provide very expressive modeling of systems, possibly with code generation. The motivation for model-driven development is to describe a system on a higher level of abstraction. This is usually done in UML and other languages by diagrams modeling specific aspects or views of a system.

MDE techniques are proposed to reduce the severity of IoT applications' development. In such an approach, applications are specied using high-level and abstract mode ls and then given as input to code generators which produce low-level code as output. For instance, PervML [15] allows developers to specify pervasive systems at a high-level of abstraction through a set of models (in UML). Nevertheless, such approaches typically require expertise in the modeling language which stakeholders might not be willing to acquire.

Ciccozzi and Spalazzese introduced MDE4IoT [2], a Model-Driven Engineering Framework supporting the modelling of Things and self-adaptation of Emergent Configurations of connected systems in the IoT. As IoT system consist of several devices, there could emerge some failures in performance of the overall system because of some non-responding devices. The article considers that in such cases the system should adapt to work without the need for these devices, and reinstall and maintain its activities. Moreover, MDE4IoT is meant to exploit the combination of a set of domain-specific modelling languages to achieve separation of concerns.

UML designer [12] is another domain-specific modeling framework built on Sirius and EMF. In UML designer, architecture models may describe the logical role of classes by class diagrams, system-actor interactions by use case diagrams, logical component models by component diagrams as well as deployment diagrams, which show the mapping of components to physical entities. Behavior can be described by sequence diagrams, or by state machines and activity diagrams. Activity diagrams describe the actions and object and control flows between actions. State machine diagrams are used in several embedded domains to model the behavior of specific objects e.g. the discrete behavior of components, in a model-driven environment, is usually defined through finite state machines.

The research presented in [3] takes advantage of the MDE principles to build a holistic development methodology involving a common, semantically expressive abstraction model, to specify a smart space with its specific services. It proposes the Resource-Oriented and Ontology-Driven Development (ROOD) methodology, which improves traditional MDE-based tools through semantic technologies for rapid prototyping of smart spaces according to the IoT paradigm. In the framework of ROOD, the Smart Space Modeling Language (SsML) was developed based on UML, that defines a Domain Specific Model (DSL). It can be used for describing high-level behaviors, interactions and context information of the entire smart space. It further defines the processing aspects related to the sensing and actuating capabilities of the smart objects, as well as the context model they manage; moreover, encapsulate these concepts into RESTful resources. The ROOD approach is realized using Obeo Designer [4].

Patel et. al presented a multi-stage model driven approach for IoT application development, based on an identification of the skills and responsibilities of the various stakeholders involved in the process [13]. Noteworthy in their approach is the use of customizable modeling languages that are customized for a particular stakeholder task and application area, where abstractions available to one stakeholder are generated from information provided by other stakeholders at previous stages. The approach is complemented by methods for generating code and mapping tasks that lead to the deployment of node-level code on composite devices.

4 Research Objectives

As explained so far, there are already several domain-specific MDE notations and tools that can be reused for designing, modeling and developing IoT-based smart system architectures and applications. They can be distinguished by different design aspects and perspectives such as views, viewpoints, components, communication protocols, etc. For the sake of interactivity by collaborative development, maintenance and consistency, evolution and flexibility, they mostly lack collaborative development support by sharing the artifacts of smart system architectures, models and applications among collaborators.

The core research objectives of this proposal are threefold:

- **Studying The State of Art.** This research initially indends to study the state of the art in model-driven IoT approaches in order to identify the best domain-specific MDE tool candidates [5], [3] dedicated especially to develop the IoT-based smart systems. The most existing MDE approaches and tools for IoT-based software systems employ standard UML profiles developed on the top of EMF [16]. As long as these tools are open-source and their underlying UML meta-models of these approaches can be reused and extended for collaborative modeling and architecture development for the IoT-based smart systems. Furthermore, this research reviews the existing collaborative MDE approaches for IoT systems in order to identify research challenges in the field.

- **Research.** The most important, challenging and long-running part of this research focuses on extended research in collaborative MDE approaches for IoT focusing on the smart technologies. In this phase, the existing collaborative modeling approaches will be studied making a operation-based textual difference representation language [11] as a potential candidate. According to [11], difference representation language is considered to be generic and as one of the efficient approach for collaborative modeling covering all aspects of collaborative modeling.
- **Validation.** This research further focuses on applying a proposed approach to real-world applications as the proof of concept.

5 Proposed Approach

There are several domain-specific MDE approaches and tools that can be reused for developing IoT-based architectures and applications. For the sake of simplicity, large-scale smart system architectures are strictly divided into parts and developed by several developers. maintenance and consistency, evolution and flexibility, they mostly lack collaborative designing and development support by sharing the artifacts of smart city architectures, models and applications among collaborators. The most existing tools use standard UML profiles by defining their own notations. However, these tools are developed on the top of EMF – eclipse modeling framework, Sirius and Obeo Designer using Ecore-based meta-modeling feature and standard UML meta-models. The collaborative modeling infrastructure introduced in Section 3 will be applied to these existing tools using their underlying meta-models.

A meta-model and modeling tool generic as well as flexible collaborative MDE infrastructure was introduced in [4]. The overall approach consists of a three-layer architecture namely: language generation, service orchestration and applications. In order to use the collaborative MDE infrastructure in any modeling domain, a difference language has to be generated for each domain. After generating difference languages, other underlying services can be orchestrated for building collaborative MDE applications for each domain language. These techniques are also applicable to the collaborative MDE of smart city applications. Language Generation. The collaborative modeling approach takes advantage of modeling deltas [4] as difference documents for storing and exchanging model changes among collaborators. Model changes in modeling deltas are represented by a textual, operation-based Difference Language (DL). Formally, DL is a family of domain-specific languages. Specific DLs for domain-specific modeling languages can be generated by DL Generator, importing the meta-models of modeling languages. For instance, the approach is applied to UML class diagrams by importing the UML class diagram meta-model in [4]. As identified in Section 2, the most existing MDE tools are developed on the top of EMF – eclipse modeling framework using Ecore-based meta-modeling feature. DL will be applied to the existing MDE tools (e.g. currently, to UML designer, to ThingML which is an EMF- and Sirius-based [10] domain-specific modeling tool [9] and to SsML in future). Specific DSL will be derived from the EMF-based Ecore meta-model [8] based on the standard UML profiles. Service Orchestration. In order to embed the collaborative MDE support behind the existing MDE tools for smart city architectures, there is a need for several operational services to perform certain collaborative modeling operations. These operations might, for instance, be calculating modeling deltas by listening for changes or comparing subsequent revisions, applying or propagating modeling deltas on models, etc. The collaborative MDE infrastructure [4] further provides a catalog of supplementary services that can recognize the DL syntax as well as manipulate and reuse DL-based modeling deltas. After generating a specific DL for a domain-specific modeling language, these services can be orchestrated in order to perform certain operations in collaborative MDE of smart city architectures. In the following, these services and their orchestrations are utilized in collaborative modeling support for smart city architectures. Applications. In the collaborative development of smart city architectures, the overall infrastructure of collaborative MDE has to provide two main scenarios: (1) interactivity of collaborators; (2) consistency of centralized artifact repositories. On the one hand, providing interactivity among collaborators is a main concern. Collaborators may use domain-specific MDE tools for working on their local workspaces. But, there must be a support for joining/opening centralized and shared smart city architectures and working on it in parallel with other collaborators. Simultaneously, the changes, they make on their copies of model, should be synchronized with other parallel instances in real-time, which is referred as interactivity. On the other hand, smart city architectures and models have to be stored in the centralized and persistent repositories. It allows for storing the histories of architectures and models under development and evolution. Repositories can then store and persist models and their histories safely for further reuse and manipulations. The collaborative MDE infrastructure [4] introduces two main scenarios of collaborative MDE namely micro-versioning and macro-versioning. Micro-versioning together with domain-specific MDE tools enables interactivity of several designers and developers (by concurrent collaboration) on the shared and centralized model-driven smart city architectures. Macro-versioning provides the consistency of model-driven smart city architectures. It is a centralized modeling delta repository with model management features such as opening working copies of models, reverting their revisions, storing complete model and their revisions, etc.

6 Expected Benefits

The central theme in this paper is to contribute towards a needs-based improvement of MDE techniques, methods, and tools, and researching novel, smart modeling techniques and applications, to address challenges posed by future software-intensive systems in the framework of IoT. Some of the most important challenges include the following:

Interactivity: The interconnected world enables global software engineering, with developer teams being distributed over long distances already becoming a common practice. For incremental and agile model-driven development and engineering, this highlights a need for truly model-based collaborative work, and tools for collaborative modeling. The micro-versioning is to be considered as a main foundation for providing interactivity support treating a centralized model as a single point of truth. Synchronization of changes will be eased by exchanging small DL-based modeling deltas. The various model designing tools (e.g. UML Designer, ThingML, SsML, etc.) which are running on different platforms can communicate with each other in terms of DL. Thereby, DL-based collaborative modeling environment serves as a common underlying infrastructure for all.

Consistency: Both the heterogeneity and the high complexity of cyber-physical systems make integrated views and models of diverse development artifacts and their interrelationships indispensable. Model consistency and integration needs to occur both throughout evolutionary life-cycle of the systems/models under development. With collaborative modeling, the histories of evolving software models are consistently preserved in modeling delta repositories by macro-versioning for further reuse and analysis. The DL syntax fully satisfies MDE concepts and provides consistency of modeling deltas as well as models under development and evolution.

Flexibility: IoT applications are dynamic, fast-evolving, and based on heterogeneous parts, e.g. realized as micro-services. This raises a need for a technology-agnostic approaches to integrate diverse subsystems flexibly. MDE can be applied for model-driven systems integration, bridging the gap between service and implementation layers using model transformations, while preserving separation of concerns. DL does not rely on any specific underlying implementation or technical space being fully independent underlying problem domain. The DL-based modeling deltas may form MDE artifacts confirming to a given meta-model, yet consist of whole system or represent changes. DL services are also realized based on service-oriented paradigm, the underlying DL and its services can easily be adapted and extended according to new changes or requirements which arise from the IoT application domain.

7 Conclusion

This paper has, first of all, shortly reviewed the state of the art in MDE approaches for IoT applications. The IoT paradigm is also subject to constant changes such as extensions, improvements and optimizations during their initial development and evolutionary life-cycle. These ever growing applications are developed and maintained by the teams of developers and designers. In order to provide team work on such huge and evolving model-driven applications, these is a need for collaborative modeling support.

This paper has demonstrated promising vision and prospects to applying collaborative modeling infrastructure [4] to engineering and developing the IoT applications that are core concepts in developing smart cities. As long as there are already sufficient model-driven engineering and development tools for IoT, development of such tools is out of the scope in this paper.

As a future work as explained in Section 3, the collaborative modeling approach will be applied to existing domain-specific modeling tools for IoT. Eventually, interactivity of developers, consistency of software projects and collaborative development will be achieved.

References

[1] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[2] F. Ciccozzi and R. Spalazzese. Mde4iot: supporting the internet of things with model-driven engineering. In *International Symposium on Intelligent and Distributed Computing*, pages 67–76. Springer, 2016.

[3] I. Corredor, A. M. Bernardos, J. Iglesias, and J. R. Casar. Model-driven methodology for rapid deployment of smart spaces based on resource-oriented architectures. *Sensors*, 12(7):9286–9335, 2012.

[4] O. Designer. Domain specific modeling for software architects, 2016.

[5] F. Fleurey, B. Morin, A. Solberg, and O. Barais. Mde to manage communications with and between resource-constrained systems. In *International Conference on Model Driven Engineering Languages and Systems*, pages 349–363. Springer, 2011.

[6] D. Giusto, A. Iera, G. Morabito, and L. Atzori. *The internet of things: 20th Tyrrhenian workshop on digital communications*. Springer Science & Business Media, 2010.

[7] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

[8] D. Guinard, V. Trifa, and E. Wilde. A resource oriented architecture for the web of things. In *Internet of Things (IOT), 2010*, pages 1–8. IEEE, 2010.

[9] R. Kateule and A. Winter. Architectural design of sensor based environmental information systems for maintainability. In *Nachhaltige Betriebliche Umweltinformationssysteme*, pages 87–96. Springer, 2018.

[10] A. G. Kleppe, J. B. Warmer, and W. Bast. *MDA explained: the model driven architecture: practice and promise*. Addison-Wesley Professional, 2003.

[11] D. Kuryazov, C. Schöneberg, and A. Winter. TOWARDS COLLABORATIVE SMART CITY MODELING, 2018.

[12] O. Network. Uml designer.

[13] P. Patel and D. Cassou. Enabling high-level application development for the internet of things. *Journal of Systems and Software*, 103:62–84, 2015.

[14] J. Rumbaugh, I. Jacobson, and G. Booch. *Unified modeling language reference manual*. Pearson Higher Education, 2004.

[15] E. Serral, P. Valderas, and V. Pelechano. Towards the model driven development of context-aware pervasive systems. *Pervasive and Mobile Computing*, 6(2):254–280, 2010.

[16] D. Steinberg, F. Budinsky, E. Merks, and M. Paternostro. *EMF: eclipse modeling framework*. Pearson Education, 2008.

[17] K. Thramboulidis and F. Christoulakis. Uml4iot—a uml-based approach to exploit iot in cyber-physical manufacturing systems. *Computers in Industry*, 82:259–272, 2016.

[18] R. H. Weber and R. Weber. *Internet of things*, volume 12. Springer, 2010.

[19] C. Yin, Z. Xiong, H. Chen, J. Wang, D. Cooper, and B. David. A literature survey on smart cities. *Science China Information Sciences*, 58(10):1–18, 2015.