# Research Proposal: Collaborative Modeling for IoT

Muzaffar Artikov

Software Engineering Department
Urgench branch of Tashkent University of Information Technologies

April 17, 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 [12].

Smart Cities are cities that employ sensors and actors to increase the efficiency of city services and processes, including environmental 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 [12]. **Model-Driven Engineering (MDE)** provides fitting foundations and is considered as an enabling technology for advancing Smart X applications.

According to aforementioned 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 [10] 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 [6]. 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 [11].

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* [12]. 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 [6], [7], [10]. 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 [10] 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 [10]. 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

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 the IoT-based smart architectures, 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 model-driven architectures for smart systems. This section shortly reviews these existing model-driven approaches and methodologies for developing IoT applications using MDE.

MDE techniques are proposed to reduce the severity of IoT applications' development. In such an approach, applications are specified using high-level abstractions using models. These models are then used to produce deployable source code. For instance, PervML [15] enables developers to specify their software architectures at abstraction levels through a set of models (in UML).

Ciccozzi and Spalazzese introduced MDE4IoT [2], a Model-Driven Engineering Framework supporting the modeling of Things and self-adaptation of Emergent Configurations of connected systems in the IoT-based smart systems. As the IoT systems consist of several connected software services and hardware components, there might be possible 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 and sustain without a need for these inactive devices, and re-install and maintain its activities. In order to avoid such failures, MDE4IoT is meant to exploit the combination of a set of domain-specific modeling languages to achieve separation of concerns.

The research presented in [3] uses 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. [13] presents a multi-stage model-driven approach for IoT application development, based on identification of the skills and responsibilities of the various stakeholders involved in the process. The approach uses configurable modeling languages that are customized for a particular stakeholder task and application area, where abstractions available to a specific 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.

ThingML [6] is another domain-specific modeling framework built on Sirius. In ThingML, the state machine diagrams are used in several embedded domains to model the behavior of specific objects e.g. the discrete behavior of components. In the MDE paradigm of ThingML, the states of hardware components are managed by defining finite state machines.

# 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 [6], [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 [12] as a potential candidate.

According to [12], 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. The IoT-based smart system architectures are very large software platforms which can be developing by different developers. Thus, for the sake of simplicity and to achieve effective results, the large-scale smart system architectures are strictly divided into parts and developed by several developers. After their initial deployment, they further have to be maintained, consisted, evolved, yet remain flexible and sustainable. These efficient results can be achieved using MDE trends and collaborative development support by sharing the artifacts of smart system architectures and models.

The most existing MDE approaches and tools use standard UML profiles and meta-models as their underlying modeling concepts. However, support for collaborative MDE for IoT is still not well developed. A meta-model generic, textual difference representation language for collaborative modeling was introduced in [12]. As long as the approach is meta-model generic, the same approach can be applied to collaborative modeling for the IoT-based smart system architectures. The approach will be applied to the existing tools using their underlying meta-models.

A meta-model and tool generic as well as flexible and sustainable [9] collaborative MDE infrastructure introduced in [12] consists of a three-layer architecture:

- **Languages.** [12] proposes a difference language (DL) for representing model changes in modeling deltas and develop the collaborative modeling approaches on the top. Model changes in modeling deltas are represented by a textual, operation-based DL which is a family of domain-specific languages. Specific DLs for domain-specific modeling languages are then generated by a DL generator service, importing the meta-models of modeling languages. The most existing MDE tools for the IoT-based smart systems are developed using standard UML profiles. In the framework of this proposal, a new DL will be extended and generated to apply to the model-driven IoT systems.

- **Services.** DL further supports a catalog of flexible and sustainable services for extending its application areas. In order to apply DL to model-driven IoT systems and collaborative modeling approach for IoT-based smart systems, supported list of DL services will be adapted and extended accordingly.

- **Applications.** After generating required DL and adapting its services, collaborative modeling use case will be realized by specific orchestrations of adapted DL services. Like in [12], this proposal also suggests two main scenarios: (1) interactivity of collaborators; (2) consistency of centralized artifact repositories. The collaborative modeling approach in [12] introduces two main scenarios of collaborative MDE: micro-versioning and macro-versioning. Micro-versioning supports the interactivity of several developers on developing model-driven IoT architectures and models. Macro-versioning supports the consistency of model-driven IoT architectures.

# 6 Expected Benefits

In the course of this research, several benefits are expected to be achieved. This research starts by investigating the existing MDE approaches for IoT-based smart systems and collaboration support for them. The state of the art will be studied and documented for further research. The core research aims to bring several advantages to the field of model-driven IoT with collaborative modeling support.

- **Interactivity:** Large-scale model-driven architectures for IoT systems are developed by the teams of developers. These develops are distributed over long distances. For incremental and agile model-driven IoT development and engineering there is a need for truly collaborative work. By the real-time synchronization of change states, interactivity of developers can achieved which sees the centralized model-driven architectures as a single point of truth. The synchronization of the changes will be facilitated by the exchange of small DL-based modeling deltas [12]. Various model design tools (e.g., PervML, ThingML, SsML) can interact with each other in terms of DL.

- **Consistency:** During the evolutionary life-cycle, model-driven IoT architectures result in several revisions that must be consisted and maintained for long-term use. With collaborative development of model-driven IoT architectures, the histories of evolving architectures are consistently preserved in repositories by macro-versioning for further reuse and analysis.

The most importantly, DL will be a common underlying language for supporting both interactivity and consistency.

While doing research, the results of research activities will be published and discussed among scientists and researchers in the field.

## 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, there is a need for collaborative modeling support.

As noted, MDE offers great opportunities for development and support for Smart X applications based on IoT. The collaborative modeling approach will accelerate the process of developing Smart X applications, allowing it to qualitatively complete and further expand at the next evolution stages.

This paper has demonstrated promising vision and prospects to applying collaborative modeling infrastructure [12] 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 5, 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, and A. Solberg. A model-driven approach to develop adaptive firmwares. In *Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, pages 168–177. ACM, 2011.

[6] 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.

[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] J. Jelschen, C. Küpker, A. Winter, A. Sandau, B. Wagner vom Berg, and J. Marx Gómez. Towards a sustainable software architecture for the nemo mobility platform. In V. Wohlgemuth, F. Fuchs-Kittowski, and J. Wittmann, editors, *Environmental Informatics – Stability, Continuity, Innovation*, page 41–48, Herzogenrath, 09 2016. Shaker. ISBN: 978-3-8440-4687-8.

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

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

[12] D. Kuryazov, A. Winter, and R. Reussner. Collaborative modeling enabled by version control. *Modellierung 2018*, 2018.

[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.