

MDD4IoT: Model-Driven Development Framework Proposal for Internet of Things

Rafael Figueira Goncalves¹, André Menolli²

¹Universidade Estadual de Londrina - UEL
<http://uel.br/>

²Universidade Estadual do Norte do Paraná - UENP
<https://uenp.edu.br/>

rafael.goncalves@uel.br, menolli@uenp.edu.br

Abstract. *The creation of Internet of Things (IoT) solutions is fairly complex, having to integrate and communicate between sensors, devices, and larger systems. Specifically, Cyber-Physical devices often lack software quality, resulting in hard-to-maintain solutions, presenting deficiencies in important aspects of IoT, such as portability and security. Software quality is of great importance to a solution's long-term success, this work describes the proposal of a framework that aims to aid in the developing of these solutions, applying a Model-Driven inspired methodology.*

Resumo. *A criação de soluções para Internet das Coisas (IoT) se trata de um processo complexo, integrando e conectando diversos sensores e dispositivos com sistemas maiores. Especificamente dispositivos Cyber-Físicos frequentemente apresentam deficiências em aspectos importantes da IoT, como portabilidade e segurança. Qualidade de software é de grande importância para o sucesso desta soluções, este trabalho descreve a proposta de um framework que tem como objetivo auxiliar no desenvolvimento destas soluções, aplicando uma metodologia Dirigida a Modelo.*

1. Introduction and Problem Characterization

The percentage of internet-connected individuals grew 1125% in 19 years [InternetWorldStats 2019], and according to [Howell 2017], the number of connected devices (not considering personal computers and smartphones) is going to reach 125 billion by 2030 (from 27 billion in 2017). Internet of Things (IoT) researches and implementations have grown exponentially, with the objective of bettering personal comfort in everyday lives with the use of an ever-growing number and types of connected devices.

Software development for these new solutions are not easily approachable, they are intrinsically complexes when compared to better established paradigms such as desktop or mobile, having higher requirements for availability, speed, as well as other challenges. For example, the low computational power in the node devices, the high vulnerability for security issues [Alaba et al. 2017]. The fundamental need for integrating with other services, and a previously unmatched concern with energy consumption only present more challenges to the systems.

The higher complexity of these solutions has a direct effect on quality, which is an important concern and a fundamental characteristic for IoT. In their work [Motta et al. 2018] has found quality to be the most agreed on IoT concern between the 3 groups pooled, Literature Review, Practitioners and Governmental Reports.

Due to the lack of research present on Cyber-Physical devices' quality, this work targets precisely this definition as its scope, also defined as [Atzori et al. 2010]'s "Things-oriented" vision, including the implementation of smart devices, items, and modules, excluding infrastructure or semantic matters.

A concept that can aid developers in maintaining high software quality is Model-Driven Development (MDD), a software development methodology that produces low-level working programming code from abstract models, facilitating compliance with standards. Its function is basically to transform a software model into executable code, either fully or partially editable, varying according to the specific demands, decreasing the need for domain knowledge for development, and avoiding the production of errors.

There are many reasons to which one can be lead to believe that MDD is especially well suited for addressing these software quality challenges in IoT development. As described by [Nguyen et al. 2015] MDD presents itself as a great way to reduce complexity, especially in IoT solutions, for its inherently modular development, which is very susceptible to code reuse and recycling. The development of an easy to use tool for applying MDD to IoT is justified by a clear lack of its existence, as stated in the conclusion of [Kapteijns et al. 2009].

There exists a clear lack of standardization for IoT development, even the definition of an IoT device itself is fairly vague, [Coetzee and Eksteen 2011] lists standardization as one out of three key areas of deficiency in IoT.

2. Objectives and Contributions

As stated in this introduction, we address software quality, one of the biggest concerns of IoT, by presenting a Model-Driven Development framework, capable of aiding the production and maintenance of high quality cyber-physical systems. In order to achieve the stated main objective, the following specific objectives are presented:

1. Develop a base approach for supporting the proposed framework
2. Develop a software framework for cyber-physical devices.
3. Develop a prototype software implementation capable of applying the developed framework.
4. Execute experiments in order to validate the results achieved with the developed software.

3. Background

The most important concept for this work is the Internet of Things, more specifically the cyber-physical devices. [Asghar et al. 2015] defines IoT as an extension of the internet in the real world, by connected objects. [Lee 2008] describes cyber-physical systems (CPS) as having computational and physical capabilities that can interact with humans and or other machines by many new modalities, our work targets the end node devices, objects or things. IoT has a very low barrier to entry, partially due to highly popular, open-source

prototyping platforms, such as Arduino [Javed 2016], which is also used as our testing platform.

Model-Driven Development is another highly important concept for this work, it allows for editable source-code generation, starting from highly abstract models, reducing the distance between solution and implementation. MDD is organized with models at three different levels, the first one is the Computation-Independent Model (CIM) which might include only domain and business models, Platform-Independent Model (PIM) which [Ayed et al. 2007] define by not having any technical details, being capable of modeling the system’s functionality in any platform, Lastly Platform-Specific Models (PSM) has the technicalities needed for deployment. These three modeling levels suffer transformations between them and in the end into the final working code.

3.1. Related Works

In the last few years, many works have addressed IoT development with MDD Methodology, many of these however are methodological only research, not yet applicable with tools, such as [Ciccozzi et al. 2017]’s MDE4IoT. The author defines Model-Driven Engineering (MDD) as a key enabling agent when the topic is IoT, the author proposes an approach that focuses on the auto-adaptation of systems and devices present in IoT. Framework for Sensor Application Development (FRASAD) was proposed by [Nguyen et al. 2015] with the intent of creating a framework capable of cope with the heterogeneity and complexity of IoT’s sensors and systems. The relevant difference to this work is it having a proprietary textual language with boilerplate code for decision making. Currently, the framework supports only models originating from a DSL meta-model, the author states the availability of his software as an Eclipse IDE plugin, but to the date of this publication, it has not been found.

4. Methodology and Current State

In this section it is presented the structure of the research followed the methodology, the work is divided into ten steps, as illustrated in Fig. 1, where a solid line indicates strict flow and a broken line indicates an optional/referential flow, each stage is defined and explained. Table 1 shows the current progress and future schedule of the research.

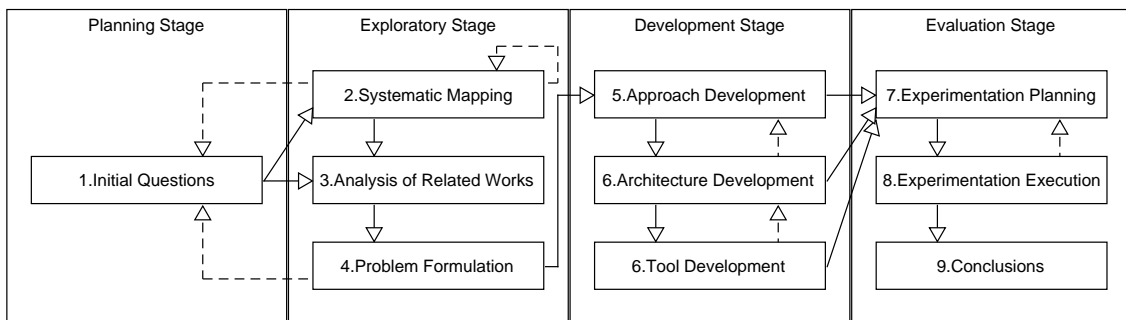


Figure 1. Research structure.

1. Initial Questions - After settled for an interest area, initial questions were established aiming to guide the research work, such questions as, What framework, standards, platform, and guidelines are being used in development? What is being used for aiding/

maintaining software quality? What are the metrics/ criteria being used in judging software quality? All these questions are being asked in the IoT/cyber-physical development area. These questions are used as an initial guide and being susceptible to changes, they were needed for the research scope setting.

2. Systematic Mapping - A systematic mapping using the guidelines by [Kitchenham et al. 2010], was conducted as the entry point to the Exploratory Stage. The initial results were partially unexpected, for example despite the number of relevant studies being large and increasing every year, from our collection none of the studies stated the software metrics or tools used in analyzing software quality, nor the programming paradigm applied. It was decided to update the systematic mapping before publishing, the only changes being of broadening the search, adding keywords to the search string, and one extra source to the databases used.

3. Analysis of Related Works - The ongoing process of analyzing related works has resulted in a sizable collection of studies, none of them being fully equivalent to the proposed approach. No-code or Low-code solutions addressing IoT are new and fail to provide adaptability in professional settings. Attributable to the applicability of MDD to highly modularized paradigms such as IoT, there exists plenty of works implementing the methodology in the IoT scope. However, none of them has a hardware-based model and produces bi-directional editable source code. This analysis only enforces [Kapteijns et al. 2009] conclusion on lacking simple to use tools contributing to MDD not delivering on its promises.

4. Problem Formulation - The initial questions aided in setting the scope of the work, the Systematic Mapping provided a clear vision of the area (to be refined as stated in step 2).

5. Approach Development - The proposed approach was developed after years of researching MDD, it presents itself as a mixture of OMG's PIM and PSM. However, it has unprecedented elements, such as the abstraction derived directly from hardware components, and source code derived from libraries. From our model displayed in Fig 2, we can observe that MicroController is populated from the selected Arduino model, from its metadata the model will know how many digital and analog ports the Arduino has, setup and loop are the only methods required for the very simple Arduino Lifecycle.

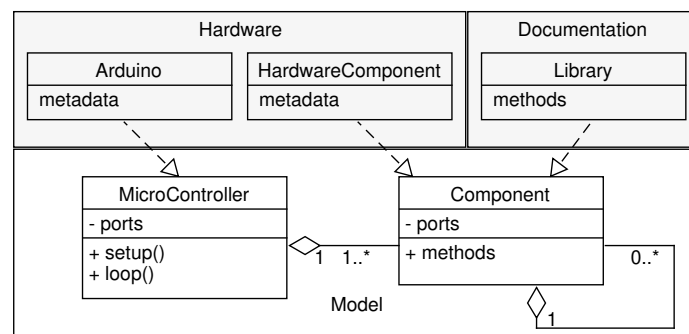


Figure 2. The hardware-based model

References

- Alaba, F. A., Othman, M., Hashem, I. A. T., and Alotaibi, F. (2017). Internet of Things security: A survey. *Journal of Network and Computer Applications*, 88(March):10–28.
- Asghar, M. H., Negi, A., and Mohammadzadeh, N. (2015). Principle application and vision in internet of things (iot). *International Conference on Computing, Communication and Automation, ICCCA 2015*.
- Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, 54(15):2787–2805.
- Ayed, D., Delanote, D., and Berbers, Y. (2007). Mdd approach for the development of context-aware applications. In *International and Interdisciplinary Conference on Modeling and Using Context*, pages 15–28. Springer.
- Ciccozzi, F., Crnkovic, I., Di Ruscio, D., Malavolta, I., Pelliccione, P., and Spalazzese, R. (2017). Model-driven engineering for mission-critical iot systems. *IEEE software*, 34(1):46–53.
- Coetzee, L. and Eksteen, J. (2011). Internet of things—promise for the future? An Introduction.
- Howell, J. (2017). Number of connected iot devices will surge to 125 billion by 2030, ihs markit says.
- InternetWorldStats (2019). World internet users statistics and 2019 world population stats.
- Javed, A. (2016). *Building Arduino projects for the Internet of Things: experiments with real-world applications*. Apress.
- Kapteijns, T., Jansen, S., Brinkkemper, S., Houët, H., and Barendse, R. (2009). A comparative case study of model driven development vs traditional development: The tortoise or the hare. *From code centric to model centric software engineering: Practices, Implications and ROI*, 22.
- Kitchenham, B. A., Budgen, D., and Brereton, O. P. (2010). The value of mapping studies – A participant-observer case study.
- Lee, E. A. (2008). Cyber physical systems: Design challenges. In *2008 11th IEEE international symposium on object and component-oriented real-time distributed computing (ISORC)*. IEEE.
- Motta, R. C., de Oliveira, K. M., and Travassos, G. H. (2018). On challenges in engineering iot software systems. In *Proceedings of the XXXII Brazilian symposium on software engineering*, pages 42–51.
- Nguyen, X. T., Tran, H. T., Baraki, H., and Geihs, K. (2015). FRASAD: A framework for model-driven IoT Application Development. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pages 387–392. IEEE.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.