Papyrus for IoT – A Modeling Solution for IoT

Saadia Dhouib, Arnaud Cuccuru, Francois Le Fèvre, Shuai Li, Benoit Maggi, Ivan Paez,

Ansgar Rademarcher, Nicolas Rapin, Jeremie Tatibouet, Patrick Tessier, Sara Tucci, Sebastien Gerard

CEA, LIST, Laboratory of Model Driven Engineering for Embedded Systems

91191 Gif-sur-Yvette CEDEX, FRANCE

ABSTRACT

Internet of Things is at the heart of many core technologies and business linking many devices, sensors, actors and humans. The Smart, Safe and Security Software Development and Execution Platform (S3P) has chosen to extend Papyrus, an open source Modeling Environment to tackle specific IoT's challenges: from heterogeneous applications development and management, real-time models to supervision of critical running system. This paper gives a highlight to the current status of "*Papyrus For IoT*".

Keywords: IoT, Model driven environment, Papyrus, Model at runtime, Complex Events Processing, IoTML.

1 CONTEXT

The S3P project [1] is a collaborative Research and Development (R&D) project that is building a platform to provide software tools for the design of Internet of Things (IoT) systems. The Smart, Safe and Security Software Development and Execution Platform (S3P) has chosen to extend Papyrus [2] to create a model-driven development environment for the specification, design, deployment, and monitoring of IoT Systems. This new initiative is called Papyrus for IoT.

S3P is funded by the French Government program "Nouvelle France Industrielle". With a combined budget of 43 million euros, the S3P project consortium includes fifteen industrial organizations and seven suppliers of technologies. The CEA List Institute of the CEA Tech, with the support of both PrismTech and MicroEJ, has built the first demonstrator of the project using Papyrus.

2 CHALLENGE

The IoT [3] is a concept and a paradigm that envisions a pervasive presence in the environment of a variety of things (objects) that are able to interact with each other and cooperate with other things to create new applications and services and reach common goals. This is achieved through wireless and wired connections and unique addressing schemes. For research and development, the primary challenges are:

1. Managing heterogeneous applications, development environments, devices and communication technologies.
2. Lack of real-time models and design methods describing reliable interworking of heterogeneous systems that is, the interactions among technical, economic, social, and environmental systems.
3. Identifying and monitoring critical system elements and detecting critical overall system states in time.

3 SOLUTION: PAPYRUS FOR IOT

A model-driven development environment has been developed as the solution to tackle these challenges. Two core concepts of model-driven engineering, namely abstraction and automation, are together, the best solution to the two first challenges. Abstraction facilitates the specification and design of complex IoT systems through the activity of modeling. Automation enables managing heterogeneous technologies through automatic model transformations and code generation to specific technological platforms. In addition to these two principles, we adopted a common architectural approach for IoT empowerment: the IoT-A architecture reference model [4] which is designed to extrapolate commonalities and define an abstraction layer that is common to all IoT-related existing architectures.

To tackle the third challenge, we propose to use development-time models to supervise a running IoT system. As a consequence, development-time models become models that reflect the running system, or in other words, Models@Runtime [5]. We now use Models@Runtime to detect critical overall system states and to help make decisions on the adaptation of the running system.

4 USING PAPYRUS FOR IOT

The IoT model-driven development and supervision environment extends the model-based engineering environment, Papyrus, to specify, design, deploy and monitor an IoT system. To illustrate this environment, we use the example of a smart IoT-based home automation system.

4.1 Specify and design complex IoT systems using a lightweight IoT-A methodology

Papyrus for IoT uses an IoT-A based methodology to guide the IoT system designer during the system development. The methodology is generic and independent from any product. It consists of five steps as shown in Figure 1.

- Step 1: Design the purpose and requirements of the system using UML use case diagrams and SysML requirements diagrams in Papyrus SysML Component [6].
- Step 2: Define the process specification. The use cases of the IoT system are formally described, based on and derived from the purpose and requirements specification.
- Step 3: Define the functional architecture of the system based on the IoT domain model, provided as an extension of MARTE and SysML, aka IoTML.
- Step 4: Define the operational platform on which the functional system will be executed.

The operational view of the system contains information about computing devices (eg. Raspberry PI, Intel Edison, ST Cortex, etc.), sensors, actuators and communication protocols (eg. Wifi, Ethernet, Zigbee, etc).

- Step 5: Define deployment plans that include information on the allocation of functional blocks (step 3) to operational ones (step 4) with Papyrus Designer Component [7].

4.2 Dynamic IoT Devices interaction simulation

Papyrus Moka Component [8] provides an extensible simulation framework to simulate models described with standard languages (e.g., UML and SysML). This component will be extended / customized to define a simulation platform for IoT models. The general purpose of this work is to validate (at least at a functional level) the behavior specified with IoTML.

4.3 Automatically deploy IoT applications on embedded devices

After defining deployment plans, the Papyrus extension for IoT systems will enable automatic code generation to the following solutions.

- Prismtech's Vortex [9] which provides a solution based on the Data Distribution Service (DDS) to dynamically deploy microservices into IoT devices, discover new devices and to migrate microservices at runtime.
- MicroEJ's operating system [10]: an environment for developing and deploying application software for IoT devices, including an operating system providing the execution environment for microservices

4.4 Use Models@Runtime to supervise system states

Papyrus for IoT offers the ability to use development-time models to supervise running system states. Our first layer solution enables monitoring of micro-services states and starting and stopping them from Papyrus. Moreover, a second layer solution focuses on critical real-time properties to check them at runtime by the ARTiMon monitoring tool [11]. This Complex Events Processing coupled with Data Intelligence enables to use Models@Runtime to modify the system's behavior at runtime in response to changes within the system.

5 Conclusion

Papyrus for IoT is a modeling environment that enable to specify, design, and deploy complex IoT systems using an IoT-A lightweight methodology. It also allows you to supervise the running system using Models@Runtime technology.

Currently, Papyrus for IoT is being developed in the context of S3P R&D project and will be publicly available as an extra Papyrus feature in early 2017.

References

[1] S3P, Smart, Safe and Security Project: http://www-list.cea.fr/images/stories/medias/communiques_de_presse/2015/uk/S3P-Press-Release-EN.pdf

[2] Papyrus, https://eclipse.org/papyrus/

[3] Internet of Things: From Research & Innovation to Market Deployment. IERC-European Research Cluster on the IoT, 2014

[4] IoT-A: http://www.iot-a.eu/

[5] Models@run.time - Foundations, Applications, and Roadmaps, Nelly Bencomo, Robert B. France, Betty H. C. Cheng, Uwe Aßmann, Lecture Notes in Computer Science, Springer 2014

[6] SysML 1.4, https://www.eclipse.org/papyrus/components/sysml/

[7] Designer, https://www.eclipse.org/papyrus/components/designer/

[8] Moka, Guermazi, S., Tatibouet, J., Cuccuru, A., Seidewitz, E., Dhouib, S., Gérard, S. "Executable Modeling with fUML and Alf in Papyrus: Tooling and Experiments", Proceedings of the 1st International Workshop on Executable Modeling co-located with ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS 2015), Ottawa, Canada, 2015.

[9] PrismTech: http://www.prismtech.com/

[10] MicroEJ: http://www.prismtech.com/

[11] ARTiMon: Reactive Property Monitoring of Hybrid Systems with Aggregation. Runtime Verification 2016, LNCS 10012, pp. 447–453, 2016. Runtime Verification 2016.

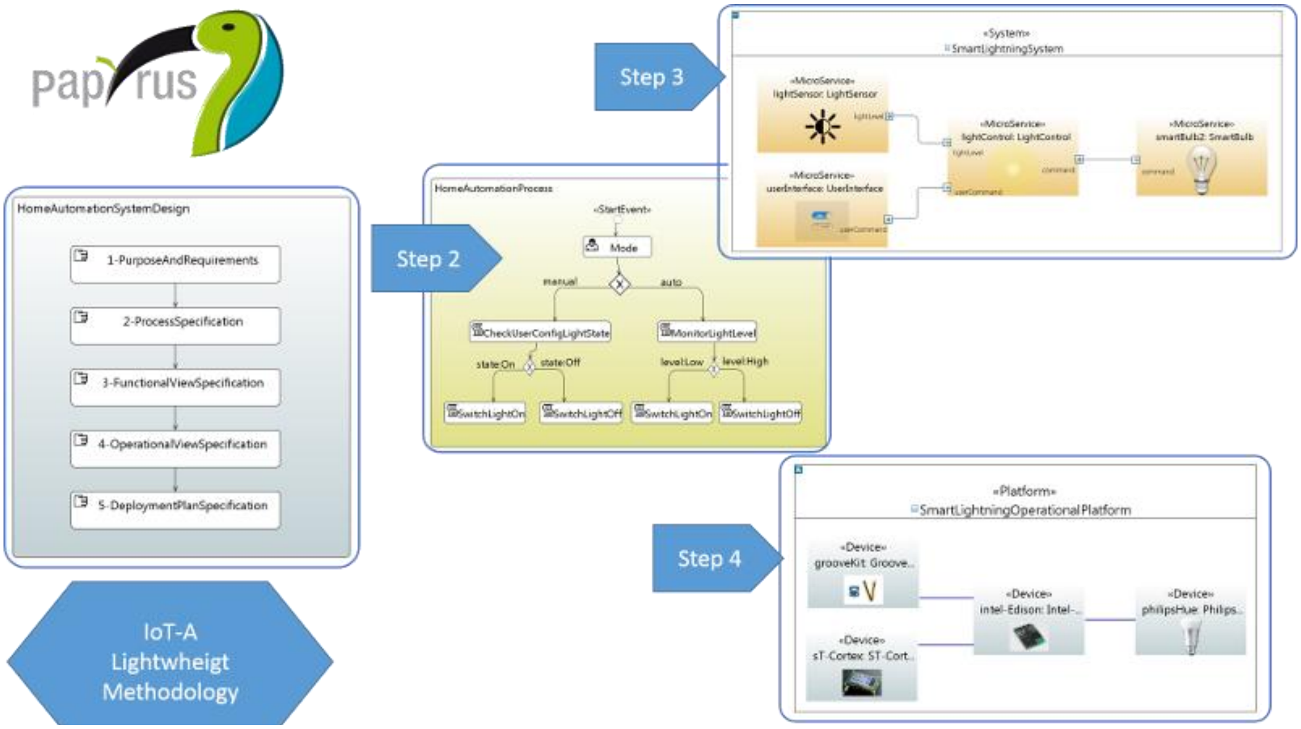

Figure 1: IoT-A lightweight methodology steps implemented in Papyrus for a first demonstrator based on a scenario where lights are controlled by temperature or lux sensors.