

---

# Models and Verification for Composition and Reconfiguration of Web of Things Applications

Gwen Salaün<sup>\*1</sup>

<sup>1</sup>Univ. Grenoble Alpes, CNRS, Grenoble INP, Inria, LIG – CNRS : UMR5217, L’Institut National de Recherche en Informatique et en Automatique (INRIA), Institut polytechnique de Grenoble (Grenoble INP), Université Grenoble Alpes, LIG – France

## Résumé

The Internet of Things (IoT) applications are built by interconnecting everyday objects over a network. These objects or devices sense the environment around them, and their network capabilities allow them to communicate with other objects to perform utilitarian tasks. One of the popular ways to build IoT applications in the consumer domain is by combining different objects using Event-Condition-Action (ECA) rules. These rules are typically in the form of IF something-happens THEN do-something. The Web of Things (WoT) are a set of standards and principles that integrate architectural styles and capabilities of web to the IoT. Even though WoT architecture coupled with ECA rules simplifies the building of IoT applications to a large extent, there are still challenges in making end-users develop advanced applications in a simple yet correct fashion due to dynamic, reactive and heterogeneous nature of IoT systems. The broad objective of this work is to leverage formal methods to provide end-users of IoT applications certain level of guarantee at design time that the designed application will behave as intended upon deployment. In this context, we propose a formal development framework based on the WoT. The objects are described using a behavioural model derived from the Thing Description specification of WoT. Then, the applications are designed not only by specifying individual ECA rules, but also by composing these rules using a composition language. The language enables users to build more expressive automation scenarios. The description of the objects and their composition are encoded in a formal specification from which the complete behaviour of the application is identified. In order to guarantee correct design of the application, this work proposes a set of generic and application-specific properties that can be validated on the complete behaviour before deployment. Further, the deployed applications may be reconfigured during their application lifecycle. The work supports reconfiguration by specifying reconfiguration properties that allow one to qualitatively compare the behaviour of the new configuration with the original configuration. The implementation of all the proposals is achieved by extending the Mozilla WebThings platform. A new set of user interfaces have been built to support the composition of rules and reconfiguration. A model transformation component which transforms WoT models to formal models and an integration with formal verification toolbox are implemented to enable automation. Finally, a deployment engine is built by extending WebThings APIs. It directs the deployment of applications and reconfigurations respecting their composition semantics.

---

<sup>\*</sup>Intervenant