Testing Uncertainty of Cyber-Physical Systems in IoT Cloud Infrastructures: Combining Model-Driven Engineering and Elastic Execution

Hong-Linh Truong
Distributed Systems Group, TU Wien, Austria
truong@dsg.tuwien.ac.at

Luca Berardinelli
Distributed Systems Group, TU Wien, Austria
berardinelli@dsg.tuwien.ac.at

ABSTRACT

Today's cyber-physical systems (CPS) span IoT and cloud-based datacenter infrastructures, which are highly heterogeneous with various types of uncertainty. Thus, testing uncertainties in these CPS is a challenging and multidisciplinary activity. We need several tools for modeling, deployment, control, and analytics to test and evaluate uncertainties for different configurations of the same CPS. In this paper, we explain why using state-of-the art model-driven engineering (MDE) and model-based testing (MBT) tools is not adequate for testing uncertainties of CPS in IoT Cloud infrastructures. We discus how to combine them with techniques for elastic execution to dynamically provision both CPS under test and testing utilities to perform tests in various IoT Cloud infrastructures.

CCS CONCEPTS

• **Computing methodologies → Model development and analysis**; • **Computer systems organization → Embedded and cyber-physical systems**; • **Software and its engineering → Software verification and validation**; *Distributed systems organizing principles*; *Software development techniques*;

KEYWORDS

testing, elasticity, uncertainty, IoT, Cloud, MDE, MBT

ACM Reference format:
Hong-Linh Truong and Luca Berardinelli. 2017. Testing Uncertainty of Cyber-Physical Systems in IoT Cloud Infrastructures: Combining Model-Driven Engineering and Elastic Execution. In *Proceedings of 2017 ACM International Workshop on Testing Embedded and Cyber-Physical Systems, Santa Barbara, CA, USA, July 13, 2017 (TECPS'17),* 4 pages.
https://doi.org/10.1145/3107091.3107093

1 INTRODUCTION

1.1 Motivation

Complex cyber-physical systems (CPS) nowadays consist of a world of physical networked *Things* interacting with cloud-based services. The CPS in our focus rely on IoT and Cloud computing as the two technological pillars. Data from *Things* pushed by sensors create high demands on cloud data storage and processing, while actuators controlling *Things* need cloud-based management services to handle diverse and large number of devices [14]. All create (and need to deal with) many uncertainties in CPS [2, 10]. Our focus is the uncertainty due to the data generated and the elasticity of IoT Cloud infrastructures [10], in addition to the uncertainties of IoT and Cloud infrastructural elements. Thus, it seems that Model-Driven Engineering (MDE), and Model-Based Testing (MBT) would be great choices to support the testing of CPS and uncertainties [2] by providing models abstracting possible configurations of IoT and Cloud infrastructural elements and their possible dynamic properties and uncertainties.

However, CPS on IoT Cloud infrastructures demand us to manage complexity throughout different activities for testing, from requirement specification, to development and deployment of both application and platform components, as well as elastic demands of these components at runtime. We can use MDE to describe CPS elements and their potential uncertainties with formal models [15] to support most engineering activities, from requirement specification to runtime analyses [4]. In this context, MDE techniques enable systematically mappings among modeling language concepts to support automatic integration.

However, for testing uncertainties of CPS in IoT Cloud infrastructures, it is not a straightforward engineering process from modeling to code transformation, to test generation and execution. Various activities need to deal with the elasticity, pay-per-use, on-demand provisioning characteristics of IoT Cloud infrastructures and software deployed atop such infrastructures. IoT and Cloud infrastructures and technologies have introduced a lot of techniques for elastic provisioning and execution of software components [1, 9]. During the execution of applications in IoT Cloud infrastructures, many operations can be performed dynamically but current MDE tools have not supported well such elastic operations.

To illustrate the above-mentioned problem, Figure 1 gives a bird's eye view of MDE and elastic execution methodologies, their relationships with generic engineering activities of CPS, including the design and operation of uncertainty testing for CPS atop IoT Cloud infrastructures. Many activities in these processes require an integrated approach between MDE and elastic execution methodologies in terms of deployment and testing activities, their input/output artifacts, the types of monitoring/testing data, and supporting tools.

TECPS'17, July 13, 2017, Santa Barbara, CA, USA

ACM ISBN 978-1-4503-5112-6/17/07…$15.00
https://doi.org/10.1145/3107091.3107093

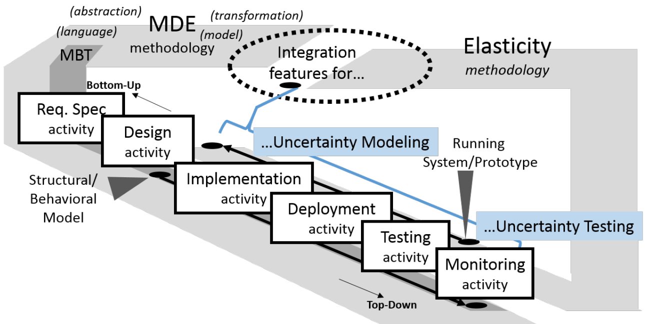

Figure 1: Combining MDE and Elasticity to support testing under uncertainty in CPS. A bird's eye view.

1.2 Contributions

In this paper, we advocate a stronger integration between MDE methodologies and elastic execution techniques towards the uncertainty testing of CPS on IoT Cloud infrastructures, focusing on integration features shown in Figure 1. We propose a tool pipeline that combines MDE approach with elastic executions of IoT Cloud tools for testing CPS. With this way, on the one hand, we leverage the abstraction power of MDE/MBT to tackle the complexity of CPS testing, and, on the other hand, we enable dynamicity and elasticity of IoT Cloud infrastructures and required testing software for CPS.
Paper Structure: Section 2 describes case studies and issues. Section 3 presents our testing tool pipeline. Section 4 summarizes related work and Section 5 concludes the paper.

2 CASE STUDIES AND ISSUES

To understand the gap between MDE and techniques for elastic execution needed for CPS uncertainty testing, let us consider the following two use cases:
Case 1 – GeoSport: this case, based on the one provided by Future Position X and Nordic Medtest in U-Test [2], aims at measuring individual ice hockey athletes through wearable devices both in indoor and outdoor environments. IoT devices attached to athletes can measure various types of data through different kinds of sensors (e.g., position, speed, and direction). Data will be relayed to a remote Central Management System (CMS), which provides live stream data or stores data for offline analytics. Various protocols can be used between sensors and CMS. For the CPS developer it is important to know (i) possible changes of sensors and their configurations, (ii) different communication protocols, and (iii) different cloud infrastructures for CMS, e.g., different data centers from different cloud providers. For testing uncertainties, we see that at the modeling level, the number of elements in the model is not huge but one needs to switch and change different concrete instances and configurations, e.g., about sensors, communication protocols, message brokers, and cloud providers, to make sure that as many as possible types of uncertainties can be discovered. For certain cases, the part of CPS under test might be fixed, e.g., sensors, but CMS can be deployed and tested for different providers (depending on the customer). In this case, data and elasticity uncertainties are crucial.
Case 2 – Monitoring and Control Equipment: this case, based on our work with industries in Vietnam, consists of various sensors used to monitor environments and equipment in smart agriculture and aquaculture, e.g., temperature, water quality, and electricity changes. The system is developed in a generic way: different sensors, configured for different application domains, are integrated into IoT gateways implemented atop Raspberry Pi. We have several gateways deployed. From gateways data is sent to the cloud through Message Queue Telemetry Transport (MQTT) protocol. In the cloud, we have a MQTT broker through which we can ingest into different cloud-based storage, like BigQuery, Cassandra and Hadoop, or ingest data into streaming engines, like Apache Apex. The reason to use various cloud back-end services is due to different types of customers (e.g., small companies or big enterprises). This also requires MQTT brokers to be configured different, such as using Google/Azure broker, cloud-based public MQTT like cloudmqtt, or private MQTT broker like Mosquitto. For this case, again data and elasticity uncertainties are crucial.

In the above cases, we face the following issues:
Issue 1 – MDE and MBT tools are not enough for testing: the application of MDE has been increasing in the CPS and IoT systems [7, 11, 15] domains. MDE provides powerful techniques for model validation, testing, verification, simulation, transformation, and execution; thus, theoretically such techniques would be very suitable for describing heterogeneous types of elements in IoT Cloud infrastructures. As the main focus of model engineers are then models, languages, transformations, they are not well-equipped with on-demand, pay-per-use system deployment and elasticity of IoT and Cloud infrastructures. In particular, MBT for IoT Cloud infrastructures requires test model(s) that (i) reflect changes in cloud service providers, software-defined capabilities of sensors, actuators and gateways, and other types of virtual resources. A survey in [6] shows that application domains in IoT Cloud environments have not been in the focus.
Issue 2 – Elastic Execution hidden from models for testing: we cannot assume that CPS (under test) atop IoT Cloud infrastructures have a fixed set of runnable components and tests. In particular, IoT Cloud infrastructures change in terms of service providers (e.g., Google Cloud Platform) or processing, storage and communication resources amounts (e.g., number of VM instances and storage space), affecting the system functionalities and their qualities. Thus, we need to incorporate elastic techniques [14] for testing. However, if elastic techniques are decoupled from MDE/MBT methodologies, it may cause lost in translation issues where test cases are automatically generated from structural and behavioral models.

To address these issues, it is important to have (i) clear MDE/MBT tools for certain activities, (ii) clear elastic execution and IoT/Cloud provisioning tools for certain activities in real systems, (iii) strong integration between these types of activities in which MDE/MBT tools must be extended or interface with IoT and Cloud techniques.

3 INTEGRATED APPROACH FOR TESTING UNCERTAINTIES

3.1 DevOps Activities for Uncertainty Testing

Before discussing how to fill the gap between MDE and elastic execution, Figure 2 outlines basic DevOps activities for development and testing of uncertainties for CPSs atop IoT Cloud infrastructures. Main DevOps activities cover:

- **Modeling**: for modeling both SUT and its uncertainties to be tested. Most effort in these types of activities can be achieved through the utilization of MDE tools.
- **Generating Test Cases**: For generating test cases and relevant artifacts. Most effort and can be achieved by using MBT tools.
- **Executing Tests**: Typically, test execution just call generated test cases, which issue calls on an existing SUT. However, in our CPS, before and during a test case, we may need to reconfigure and deploy test utilities and SUT. This activity thus cannot be done simply through typical MDE/MBT tools which are not integrated with techniques for on-demand deployment. It needs a combination of MBT-test unit and elastic execution techniques in IoT and Cloud in order to deal with various situations related to (part of) SUT and test utilities.
- **Deploying SUT**: In the cloud/IoT one can switch from one provider to another provider for certain services/IoT. Thus, in many cases, (part of) the SUT is just deployed before or during tests. Thus, here we need to incorporate techniques from elasticity and dynamic provisioning.
- **Deploying Test Utilities**: Because of multiple providers, test utilities can be deployed at different systems. Furthermore, due to elasticity, we need to reconfigure test utilities.

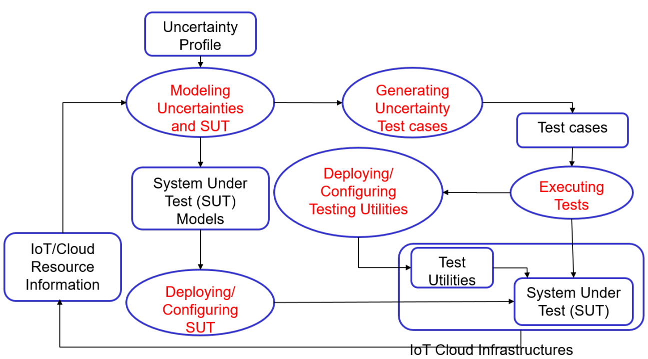

Figure 2: DevOps activities in uncertainty testing

These DevOps activities are at a high-level view but show that a strong integration is needed. Consider **Case 1** in Section 2, if we want to change the communication from sensors to the CMS at the backend, we might need to deploy the right test utilities. Consider **Case 2**, in order to test the MQTT broker deployed in a private cloud, one can deploy the MQTT broker in different configurations and tests by changing the configuration (testing the elasticity uncertainty). At the same time, we can measure data quality and test the data uncertainty. This involves the deployment and configuration of the MQTT in the SUT. In case, if we just switch from the MQTT broker in Google to Amazon, we need to deploy different test utilities and other support components into appropriate Google or Amazon resources because different cloud providers require different test utility even for the same type of MQTT brokers.

3.2 MDE and Elastic Execution – Testing Tool Pipeline

From the above-mentioned activities, to combine MDE/MBT tools with elastic execution for uncertainty testing, we propose a generic tool pipeline in Figure 3. Analogy to state-of-the-art data pipelines, we have different steps invoking different tools to model, generate, transform, deploy, execute and analyze various artifacts for testing.

In the top part of Figure 3 we show the uncertainty test pipeline of activities (squared boxes) and artifacts (slanted boxes) that are inherent to the process of DevOps for uncertainty testing. At the end of the pipeline, various test cases are executed together with elastic execution of IoT and cloud infrastructural elements. The corresponding supporting software tool categories for MDE/MBT and for elastic execution are shown below the pipeline, with tool examples listed between parenthesis. Both commercial and open source off-the-shelves tools can be adopted as is and extended via plug-ins (e.g., modeling tools integrated in Eclipse IDE) while other ones may be developed on purpose (e.g, to generate artifacts via model transformations) to create a smooth integration between MDE/MBT and elastic execution methodologies.

We then map activities with the corresponding tool categories via circled letters. For modeling (a), it is easy to see that we have a lot of suitable, open source MDE tools. Both domain specific languages (DSL, via Eclipse EMF) or general purpose one (like UML via Papyrus) can be chosen. Due to elastic execution, the integration between these tools with elastic execution techniques (c)should be in the focus and supported by model integration frameworks (b), mostly based on model query and transformation languages (e.g., OCL, ATL, xText). Especially, when we extract information and generate artifacts for testing, the key question is how we incorporate dynamic information about targeting IoT and Cloud infrastructures where resources and services are highly dynamic. In this view, models and tools can be both extended by integrating advanced resource modeling concepts (e.g., MARTE profile for UML) and IoT and Cloud information services.

Inside test cases, we need to glue various tasks for IoT and cloud resources provisioning and test utilities deployment. Such tasks also require up-to-date information about IoT and cloud resources. Here there might be different methods can be implemented: (1) generating tests and configuration cases and running them in a sequence, e.g., first deployment and second, testing, or (2) for example, if we want to test uncertainty due to elasticity, then during a test case, resources are added and removed dynamically together with components of SUT. Both require complex and smart algorithms to generate test cases and resource provisioning cases in the right way from the model. While several tools do exist, such algorithms and integration techniques have not well addressed.

Test execution will be highly distributed but we expect many existing techniques from MBT can be used (or extended with distributed test case execution models). However, as we test CPS in IoT Cloud infrastructures, in the analytics of tests, we also need to link various features about testing log transformation, data collection, etc. into the test cases. For example, when a test case is executed in a remote cloud resource, testing logs must be transferred into the right service for test analytics. In the cloud environment, it is not enough to assume that test log files will be shifted to or to assume that test cases will send data to a dedicate server. We also need to collect other information related to the underlying IoT and Cloud resources, such as virtual machines and containers. This requires us to leverage also various tools for collecting and processing various

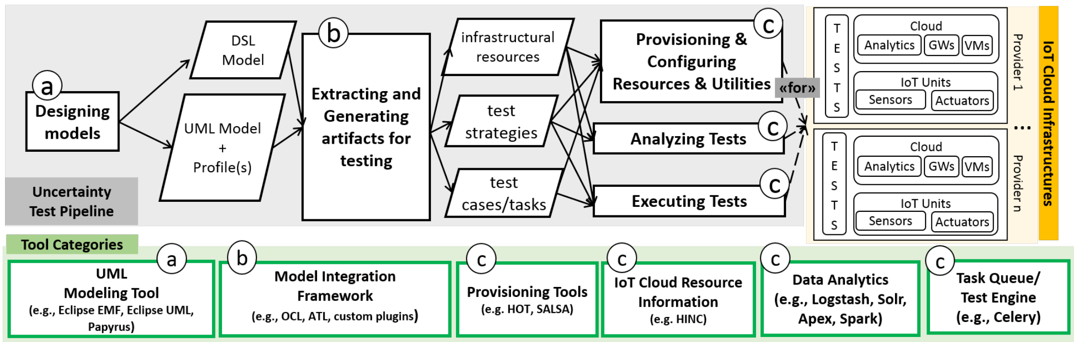

Figure 3: Uncertainty test pipeline, Tools, and IoT cloud infrastructures

types of logs relevant to testing, such as Logstash, Fluentd and Nifi, to perform on demand log transformation for data analytics. Because we rely on different data for test analytics, we might need different data processing frameworks, such as Apache Spark, Apex and Solr, to process, search and extract various types of data (including models). This is related to the use of big data techniques for test analytics.

4 RELATED WORK

In MDE/MBT many tools have supported testing and application development for CPS, IoT and Cloud. We investigate the integration aspect between MDE/MBT and elastic techniques and found that most tools just perform a straight-transformation to the underlying IoT or Cloud infrastructures. They do not support well techniques for elastic execution to deal with IoT and Cloud infrastructures. Recently [13] shows MBT for MQTT with different configurations but the integration of elastic techniques for testing/MBT.

In IoT and Cloud platforms, various tools allow us to perform provisioning of resources and execution of tasks on-demand; they are very suitable for integration with MDE. However, most MDE tools for Cloud just map model elements to cloud resources, e.g. [5, 8]. Elastic testing enables the execution of test cases in different cloud resources; they are more or less executing tasks for testing in the cloud [3]. The above-mentioned tools naturally fit into our tool pipeline but they do not focus on uncertainty testing. Work on generating test cases from models [12] so far has not well supported the interwoven elastic execution.

5 CONCLUSIONS

In this paper, we analyze activities for testing uncertainties in IoT Cloud infrastructures and applications. We show how we need to combine MDE/MBT with cloud elasticity techniques to support testing uncertainty. Our initial work is focused on design the tool pipeline. To realize and perform a proof-of-concept of our proposal in this paper, we are currently implementing our tool pipeline based on our view in Figure 3. Successful criteria for such a pipeline in our view will be based on capabilities to generate an deploy test plans and SUT working seamlessly with state-of-the-art IoT Cloud systems, to reduce the provisioning and testing cost, and to discover data and elasticity uncertainties.

ACKNOWLEDGMENT

This work was partially supported by the European Commission in terms of the U-Test H2020 project (H2020-ICT-2014-1 #645463)

REFERENCES

[1] https://aws.amazon.com/cloudformation/. (????).
[2] U-Test H2020 Project Web Site. http://www.u-test.eu/. (????).
[3] Michel Albonico, Amine Benelallam, Jean-Marie Mottu, and Gerson Sunyé. 2016. A DSL-based Approach for Elasticity Testing of Cloud Systems. In *Proceedings of the International Workshop on Domain-Specific Modeling (DSM 2016)*. ACM, New York, NY, USA, 8–14. https://doi.org/10.1145/3023147.3023149
[4] Nelly Bencomo, Robert B. France, Betty H. C. Cheng, and Uwe Aßmann (Eds.). 2014. *Models@run.time - Foundations, Applications, and Roadmaps [Dagstuhl Seminar 11481, November 27 - December 2, 2011]*. Lecture Notes in Computer Science, Vol. 8378. Springer.
[5] Amine Benelallam, Abel Gómez, Massimo Tisi, and Jordi Cabot. 2015. Distributed Model-to-model Transformation with ATL on MapReduce. In *Proceedings of the 2015 ACM SIGPLAN International Conference on Software Language Engineering (SLE 2015)*. ACM, New York, NY, USA, 37–48.
[6] Robert V. Binder, Bruno Legeard, and Anne Kramer. 2014. Model-based Testing: Where Does It Stand? *Queue* 13, 1, Article 40 (Dec. 2014), 9 pages.
[7] Federico Ciccozzi and Romina Spalazzese. 2017. *MDE4IoT: Supporting the Internet of Things with Model-Driven Engineering*. Springer International Publishing, Cham, 67–76.
[8] Michele Guerriero, Saeed Tajfar, Damian A. Tamburri, and Elisabetta Di Nitto. 2016. Towards a Model-driven Design Tool for Big Data Architectures. In *Proceedings of the 2Nd International Workshop on BIG Data Software Engineering (BIGDSE '16)*. ACM, New York, NY, USA, 37–43.
[9] Duc-Hung Le, Hong Linh Truong, Georgiana Copil, Stefan Nastic, and Schahram Dustdar. 2014. SALSA: A Framework for Dynamic Configuration of Cloud Services. In *CloudCom*. IEEE Computer Society, 146–153.
[10] Shaukat Ali Tao Yue Dipesh Pradhan Stefan Nastic Hong-Linh Truong Martin Schneider Max Bureck Christian Hein Man Zhang, Bran Selic. 2015. *Understanding Uncertainty in Cyber-Physical Systems, https://www.simula.no/file/d12pdf/download.* Technical Report. U-Test Project.
[11] Ferry Pramudianto, Indra Rusmita Indra, and Matthias Jarke. 2013. Model Driven Development for Internet of Things Application Prototyping.. In *SEKE*. 703–708.
[12] Monalisa Sarma and Rajib Mall. 2007. Automatic Test Case Generation from UML Models. In *Proceedings of the 10th International Conference on Information Technology (ICIT '07)*. IEEE Computer Society, Washington, DC, USA, 196–201.
[13] Martin Tappler, Bernhard K. Aichernig, and Roderick Bloem. 2017. Model-based testing IoT communication via active automata learning. In *10th IEEE International Conference on Software Testing, Verification and Validation,*. Tokyo, Japan.
[14] Hong Linh Truong and Schahram Dustdar. 2015. Principles for Engineering IoT Cloud Systems. *IEEE Cloud Computing* 2, 2 (2015), 68–76.
[15] Man Zhang, Bran Selic, Shaukat Ali, Tao Yue, Oscar Okariz, and Roland Norgren. 2016. Understanding Uncertainty in Cyber-Physical Systems: A Conceptual Model. In *Modelling Foundations and Applications - 12th European Conference, ECMFA 2016, Vienna, Austria, July 6-7, 2016, Proceedings*. 247–264.