Challenges & Opportunities in Low-Code Testing

Faezeh Khorram, Jean-Marie Mottu, Gerson Sunyé

LS2N, Université de Nantes, IMT Atlantique, Nantes, France

ACM/IEEE 23rd International Conference on Model Driven Engineering Languages and Systems (MODELS '20 Companion)

"This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 813884".

lowcomote

Institut Mines-Télécom

Problem Statement

- ***Low-code Testing*** required for Quality Assurance of Low-Code Systems
- A few techniques are offered by commercial LCDPs
- A new area without a precise definition
  - Lack of information on the impacts of low-code principles on testing activities
  - High dependency of existing LCDPs to technical testing tools
- Lack of academic research

No formal structure to the ideas, concepts, and research questions of this area

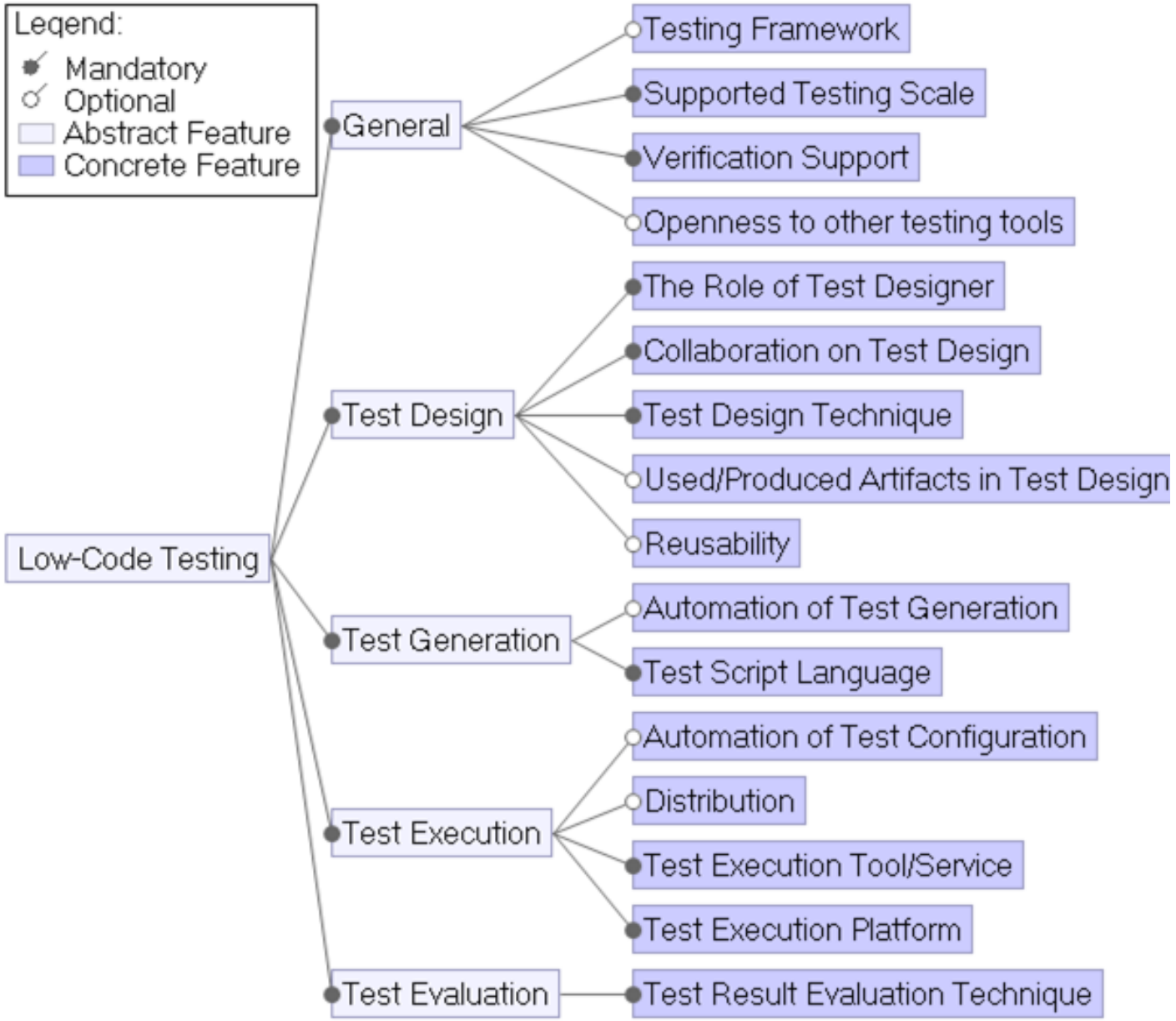
Contribution

- Studying the testing component of 5 commercial LCDPs
  - Mendix, OutSystems, PowerApps, Temenos Quantum, Lightning
- Extracting Low-code testing features
  - A feature set with possible values for them
- Characterizing existing challenges into three categories from research point of view
- Proposing opportunities for future work based on the previous attempts

Low-Code Testing Feature Set

- Characterizing Low-Code Testing
  - Comparison of the existing tools
  - Finding the gaps in the state-of-the-art
  - Help on developing new tools
  - Roadmap for future research

- Extracting 16 Features
  - Based on Low-Code principles
  - Considering capabilities and deficiencies of the existing commercial tools
  - Providing some possible values for them



Example of Feature: Test Design Technique

	MBT	Record & Replay	DDT	Keyword -driven	BDD/ TDD	AI
mendix	✓	✓	✓	✓		
outsystems	✓	✓			✓	
PowerApps		✓	✓	✓		
TEMENOS | kony		✓			✓	
salesforce lightning platform		✓	✓	✓	✓	✓

✓ Supported by Low-Code testing tool

✓ Supported by integrated third-party testing tool

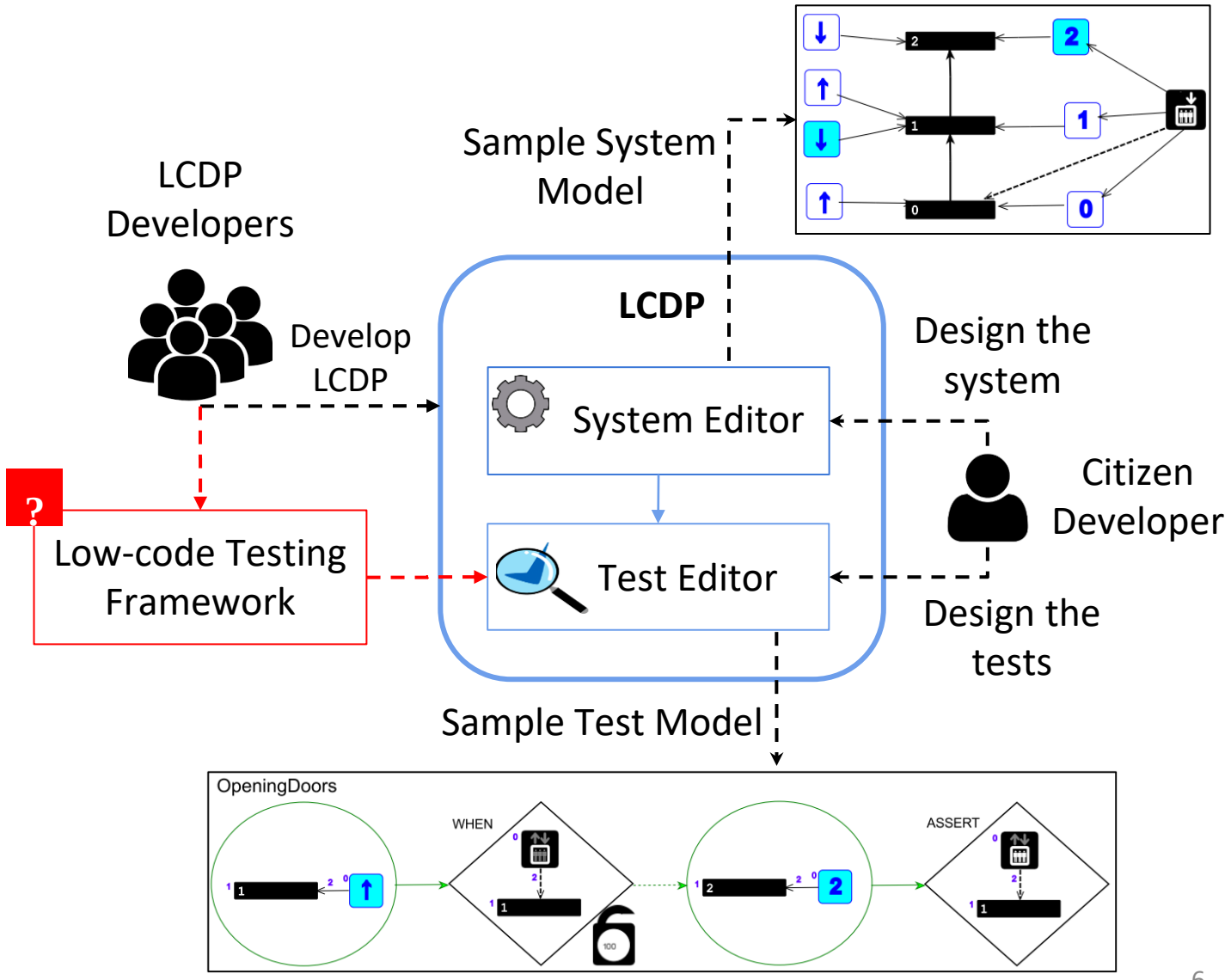
Identified Challenges & Opportunities

Main observation: Lack of Low-Code Testing Framework

- For the development of Low-code testing components
- To be Generic, Reusable, and Open to access

Three sets of challenges

- Role of citizen developer in testing
- Need for high-level test automation
- Cloud Testing



The Role of Citizen Developer in Testing

Challenges

- Low-level technical knowledge of Citizen developer
  - Simple, familiar, and non-technical approaches are required
- Modeling test cases in the same way as system modeling
  - Testing support for DSLs is required

Previous Attempts

- Automated Testing Support for Reactive Domain-Specific Languages (Meyers, et al. 2016)
  - Restricted to DSLs with rule-based semantics
  - Not efficient: The semantics must be regenerated for each test case
  - Limited support for testing elements

Opportunities

- Support for other testing DSLs (e.g., TDL)
- Proposing a generic approach for testing support of different DSLs
- Assistant chatbots and recommendation systems for helping citizen developer in test case definition

The Need for High-level Test Automation

Challenges

- High-level of automation alongside low dependency to technical knowledge
  - *Suitable techniques*: Data-Driven, Model-Based Testing (MBT), and Record and Replay
- Lack of reusability for MBT tools because of their dependency to specific DSLs
- Lack of MBT tool repository and challenges in MBT tool selection/creation

Previous Attempts

- 70 MBT supporting tools are proposed from 2006 to 2016 (Bernardino, et al. 2017)
  - For different domains (i.e., for different DSLs)

Opportunities

- Development of a repository for existing MBT tools
- Application of MBT in the testing component of LCDPs
  1. If MBT is already applied to the LCDP's underlying DSL, and associated tools exist: an appropriate tool can be selected from the pool.
  2. Otherwise: new MBT tools adapted to the new DSL should be implemented.

Cloud Testing

Challenges

- Cloud-native of LCDPs and their support for building cloud-based applications
  - Need for supporting cloud testing in LCDPs
- Supporting cloud in MBT

Previous Attempts

- MIDAS: A cloud-based MBT testing platform for Software-Oriented Architectures (SOA)
  - Dedicated to MIDAS DSL
  - Usable for automatic generation of test cases, so not applicable for test case modeling

Opportunities

- Application of MIDAS approach (i.e., MBT as a service) for different DSLs
- Proposing a cloud-based low-code testing framework
  - A comprehensive framework that auto-generates test-specific services for a given DSL

Future Work

- Definition of an executable testing language
  - Based on Test Description Language (TDL)
  - Independent and customizable for different DSLs

- Definition of a cloud-based low-code testing framework
  - Based on the proposed testing language
  - Auto generation of test-specific services for a given DSL
  - Supporting test execution in the cloud

Challenges & Opportunities in Low-Code Testing

Faezeh Khorram, Jean-Marie Mottu, Gerson Sunyé

LS2N, Université de Nantes, IMT Atlantique, Nantes, France

ACM/IEEE 23rd International Conference on Model Driven Engineering Languages and Systems (MODELS '20 Companion)

"This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 813884".

lowcomote

Institut Mines-Télécom