SYSTEMATIC MODEL-BASED AND SEARCH-BASED TESTING OF CYBER-PHYSICAL SYSTEMS

Shaukat Ali, PhD, Senior Research Scientist
Email: shaukat@simula.no

© All rights reserved
OUTLINE OF THE PRESENTATION

- Background
  - Cyber-Physical Systems
  - Search-based Software Engineering
- Results from previous projects
- Ongoing Project
- Summary, Experiences, and Lessons Learnt
BACKGROUND

Cyber-Physical Systems (CPS)
CYBER-PHYSICAL SYSTEMS

CPSs are the new generation of connected embedded systems integrating cyber-technologies, software, and physical components interacting with each other via information and physical interfaces [1].


Geo Sports: Picture courtesy Future Position X, Sweden

Handling Systems: Picture courtesy ULMA Handling Systems, Spain
Testing Levels for CPS

**Application Level**: Events and data coming from the user space, e.g., from applications and human.

**Infrastructure Level**: Events and data coming from, e.g., physical units, network equipment, and cloud infrastructure.

**Integration Level**: Interactions between Application and Infrastructure levels.
SCIENTIFIC AND TECHNICAL CHALLENGES

- Heterogeneous, Large-scale Embedded Systems
- Dealing with Novel Interactions
  - Software, Hardware, Communication, Human
- Dealing with Uncertainty
- Verifying and validation of extra-functional properties such as performance, robustness, ..
- Autonomous
WHY IS IT IMPORTANT TO TEST CPS?

- Applications, e.g., Healthcare, Aerospace, Avionics, Oil/gas and Maritime, Industrial Automation, and Tele-communication
- Current applications > $32.3 trillions. By 2025, > $82 trillions [1].
- CPSs must be dependable, i.e., safe, trustworthy, reliable, robust, ...
- Improving CPS dependability via systematic and automated testing

BACKGROUND

SEARCH-BASED SOFTWARE ENGINEERING
Search-Based Software Engineering

“Using search techniques to search large search spaces, guided by a fitness function that captures properties of the acceptable software artifacts we seek”[1]

Search Techniques: Genetic Algorithms, Particle Swarm Optimization, ...

Large Search Spaces: Millions or billions of possible solutions to search from.

Fitness Function: To determine solution $x$ is better than Solution $y$.

RESULTS FROM SELECTED PROJECTS

ROBUSTNESS TESTING OF VIDEOCONFERENCING SYSTEMS
What is Robustness?

“Robustness is the degree to which a software component functions correctly in the presence of exceptional inputs or stressful environmental conditions” (IEEE Std 610.12-1990)
MODEL-BASED ROBUSTNESS TESTING IS CHALLENGING

• Modeling robustness behavior makes modeling highly complex and redundant

• Automated generation of executable test cases from robustness models
  ✓ Targeted to reveal robustness faults
  ✓ Generating test data
  ✓ Defining appropriate test strategies for robustness testing
CASE STUDY:
VIDEOCONFERENCING SYSTEMS (CISCO)
SOLUTION FOR MODEL-BASED ROBUSTNESS TESTING
MODELLING METHODOLOGY

A1: Develop a conceptual model of a system
A2: Develop a behavioral model of the system as UML state machines
A3: Develop fault taxonomy
A4: Develop a class diagram for a robustness aspect
A5: Develop a state machine for the aspect
A6: Define ordering of aspects using a state machine
A7: Weave aspects with behavioural models

UMLStateMachines
FaultTaxonomy
AspectStateMachines
WeavingDirectiveStateMachines
WovenStateMachines
UMLClassDiagram
AspectClassDiagram

User Conference
on Advanced Automated Testing
[ simula . research laboratory ]
MODELING FUNCTIONAL BEHAVIOR
MODELING FUNCTIONAL BEHAVIOR

![State diagram showing states and transitions with conditions and actions]
MODELING ASPECT CLASS DIAGRAM

- **NftpType**
  - Reorder
    - delay : Delay
    - reorderValue : NFP_Percentage
    - correlation : NFP_Percentage
    - distribution : Distribution
    - gap : Integer

- **NftpType**
  - Delay
    - value : Integer
    - unit : TimeUnitKind
    - randomVariationValue : Integer
    - randomVariationUnit : TimeUnitKind
    - correlationValue : NFP_Percentage
    - distribution : Distribution
    - gap : Integer

- **CommunicationMedia**
  - Network
    - packetDelay : Delay
    - packetLoss : PacketLoss
    - duplicate : NFP_Percentage
    - corrupt : NFP_Percentage
    - reorder : Reorder

- **NftpType**
  - PacketLoss
    - packetLossValue : NFP_Percentage
    - correlationValue : NFP_Percentage
    - distribution : Distribution
    - gap : Integer

```
<enumeration>
Distribution

normal
pareto
paretonormal
```
MODELING ASPECT STATE MACHINES

   - Introduction

   - self.network.packetLoss.packetLossValue.value > 0
   - self.packetDelay.delay.value > 0 and self.packetDelay.delay.unit = TimeUnitKind::ms
   - self.reorder.delay.value > 0 and self.reorder.delay.unit = TimeUnitKind::ms
   - self.corrupt.value > 0
   - self.duplicate.value > 0

2. Pointcut: type: SelectionType::Subset, selectionConstraint: “name != ‘Idle’ and name != ‘PresentingWithoutCall’”

   - Introduction

   - self.network.packetLoss.packetLossValue.value = 0
   - self.packetDelay.delay.value = 0
   - self.reorder.delay.value = 0 and self.reorder.delay.unit = TimeUnitKind::ms
   - self.corrupt.value = 0
   - self.duplicate.value = 0

3. Pointcut: type: SelectionType::One, selectionConstraint: “Idle”

   - Disconnect
     - after (time)
     - DisconnectAll
WOVEN STATE MACHINE

- self.network.packetLoss.packetLossValue.value > 0
- self.packetDelay.delay.value > 0 and self.packetDelay.delay.unit = TimeUnitKind::ms
- self.reorder.delay.value > 0 and self.reorder.delay.unit = TimeUnitKind::ms
- self.corrupt.value > 0
- self.duplicate.value > 0

- self.network.packetLoss.packetLossValue.value = 0
- self.packetDelay.delay.value = 0
- self.reorder.delay.value = 0 and self.reorder.delay.unit = TimeUnitKind::ms
- self.corrupt.value = 0
- self.duplicate.value = 0
# RESULTS FROM MODELING

<table>
<thead>
<tr>
<th>Crosscutting behavior</th>
<th>Using aspects</th>
<th>Without aspects</th>
<th>Effort Saved (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>States (Added)</td>
<td>Transition (Added)</td>
<td>States (Modified/Added)</td>
</tr>
<tr>
<td>Updating audio constraints</td>
<td>1</td>
<td>-</td>
<td>86 (Modified)</td>
</tr>
<tr>
<td>Updating video constraints</td>
<td>1</td>
<td>-</td>
<td>86 (Modified)</td>
</tr>
<tr>
<td>Media quality recovery</td>
<td>3</td>
<td>3</td>
<td>20 (Added)</td>
</tr>
<tr>
<td>Network communication</td>
<td>3</td>
<td>3</td>
<td>20 (Added)</td>
</tr>
<tr>
<td>Add Guard</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
TEST CASE GENERATION

• Constraint solving using search algorithms for Test Data Generation (EsOCL tool)
  ✓ Violates properties of the environment to check robustness of the system against those violations
  ✓ Search algorithms such as GA, 1+1 (EA), ..
  ✓ EsOCL’s performance is practically applicable

• Developed a tool TRansformation-based tool for Uml-baSed Testing (TRUST)
  ✓ Supports configurable and extensible features such as input models, test models, coverage criteria, test data generation strategies, and test script languages.
  ✓ Applied to ABB Robotics and Cisco case studies
RESULTS FROM EXISTING PROJECTS

PRODUCT LINE TESTING
USING FEATURE MODELS TO SELECT AND CONFIGURE BEHAVIORAL MODELS

- Select features in FM_T through the Selection Front-end
- Configure attributes CFM_B through the Configuration Front-end
STATE MACHINE VARIABILITY
STATE MACHINE MODEL ELEMENT VARIABILITY

User Conference on Advanced Automated Testing
ATTRIBUTE VARIABILITY

User Conference on Advanced Automated Testing

context Multisite inv: self.MaxNumberOfCalls = self.MaxCalls
ATTRIBUTE VARIABILITY

```plaintext
<<Restrictions>>
<VideoCall>
  <<Restrictions>>
  Multi-way
  Multi-site

<<Restrictions>>
<VideoCall>
  ps.object: Multi-way
  ps.object: Multi-site

<<Pointcut>>
name = 'Select State Machines', type = SelectionType::All

<<Introduction>>
{context Saturn inv:
  self.networkServices.H323_Mode =
  self.networkServices.ModeOfH323}

<<Trace>>
NetworkServices
  H323_Mode : String
  ModeOfH323 : String

Saturn
```

User Conference on Advanced Automated Testing

20-22/10/2015
© All rights reserved
CONCLUDING REMARKS

• Applied to configure several products in a Videoconferencing Product Line of Cisco.
• The configured models were successfully used to generate test cases using TRUST.
RESULTS FROM SELECTED PROJECTS

ROBUSTNESS TEST SELECTION IN A MARITIME APPLICATION
CONTEXT OF THE PROBLEM

• As part of a project in Certus [1] with Kongsberg Maritime as industrial partner
• Overall optimization objective is to find a set of test cases to “break” a CPS as soon as possible

1. Certus Software Verification and Validation Center, http://certus-sfi.no/
OPTIMIZATION OBJECTIVES

• Input: **Set of test cases** focusing on testing software, hardware (mechanical, electronics, ..), interactions among them

• Cost
  ✓ Overall Execution Time,..

• Effectiveness
  ✓ Probability of Failure, Risk, Safety Level, ...

• Fitness Function using Cost and Effectiveness measures

• Existing Implementation of Search Algorithms
TOOL SUPPORT

Select test cases based on the following choices:

Execution Time: 40 hours

Priority: Efficiency priority

Probability: Efficiency probability

Consequence: Efficiency consequence

Risk: 

Context: Component Under Test

Type of Tests: Feature
CONCLUDING REMARKS

• Some preliminary results are obtained.
• The tool seems to be promising based on initial experiments.
• More case studies are being conducted.
ONGOING CPS TESTING PROJECT: H2020

OBJECTIVES OF U-TEST

- **Objective**: Improve the dependability
- **Means**: Model-based and Search-based Testing
- Objective will be achieved by:
  - ✔ Uncertainty Taxonomy
  - ✔ Holistic Modeling and Testing Frameworks
  - ✔ Standards
Case Study Providers: Geo Sports Future Position X, Sweden
Case Study Providers: Automated Warehouses
ULMA Handling Systems, Spain
CONSORTIUM

• **Research Partners**
  ✓ Simula Research Laboratory, Norway
  ✓ Fraunhofer FOKUS, Germany
  ✓ TU Wien, Austria

• **Tool Vendors**
  ✓ Easy Global Marketing
  ✓ FOKUS!MBT

• **Exploitation**
  ✓ Ikerlan

• **Test Bed Provider**
  ✓ Nordic MedTest

• **Project Management and Administration**
  ✓ Oslo Medtech
SUMMARY OF RESULTS

MODELING AND TESTING SOLUTIONS
## CPS MODELING FOR MBT

<table>
<thead>
<tr>
<th>Category</th>
<th>Objective</th>
<th>Modeling Solution</th>
<th>CPS</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robustness Testing</td>
<td>Test Case/Data Generation</td>
<td>UML Class Diagram, UML State Machine, AspectSM, and OCL</td>
<td>Video-Conferencing Systems</td>
<td>Cisco Systems, Norway, Tomra, Norway WesternGeco, Norway</td>
</tr>
<tr>
<td></td>
<td>Test Selection</td>
<td>UML Class Diagram and OCL</td>
<td>Bottle Recycling System</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Ship Navigation System</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Dynamic Positioning Systems</td>
<td>Kongsberg Maritime, Norway</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Vessel Control Systems</td>
<td></td>
</tr>
<tr>
<td>Product Line Testing</td>
<td>Test Selection, Prioritization, Minimization</td>
<td>Feature Model Component Family Model</td>
<td>Video-Conferencing Systems</td>
<td>Cisco Systems, Norway</td>
</tr>
<tr>
<td></td>
<td>Test Case/Data Generation</td>
<td>UML Class Diagram, UML State Machine, AspectSM, OCL and Feature Model</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functional Testing</td>
<td>Test Case/Data Generation</td>
<td>RUCM, RTCM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experiences

<table>
<thead>
<tr>
<th>Modeling Objective</th>
<th>Modeling Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Specification, V&amp;V</td>
<td>RUCM, RUCM4RT, RTCM, AspectRUCM, RUCM Variability, RUCM4Uncertainty</td>
</tr>
<tr>
<td>CPS Product Line Engineering</td>
<td>UML, OCL, MARTE, the SimPL Profile</td>
</tr>
<tr>
<td>Model Based Testing</td>
<td>UML, OCL, MARTE, the AspectSM Profile</td>
</tr>
<tr>
<td>Model Based Uncertainty Testing</td>
<td>UML, OCL, MARTE, SysML, the Uncertainty Profile</td>
</tr>
<tr>
<td>Model Based Product Line Testing</td>
<td>Feature Model, UML, the AspectSM Profile</td>
</tr>
</tbody>
</table>

Experiences:

- Largely relied on standards
- Proposed several profiles for various purposes
- Developed our own NL-based and model-based solutions
- Developed tools
- Intentionally made effort to reduce modeling effort
- Evaluated with controlled experiments and industrial case studies
- Used existing search algorithms and in rare cases extended
Lessons Learnt

<table>
<thead>
<tr>
<th>Modeling Objective</th>
<th>Modeling Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Specification, V&amp;V</td>
<td>RUCM, RUCM4RT, RTCM, AspectRUCM, RUCM Variability, <strong>RUCM4Uncertainty</strong></td>
</tr>
<tr>
<td>CPS Product Line Engineering</td>
<td>UML, OCL, MARTE, the SimPL Profile</td>
</tr>
<tr>
<td>Model Based Testing</td>
<td>UML, OCL, MARTE, the AspectSM Profile</td>
</tr>
<tr>
<td>Model Based Uncertainty Testing</td>
<td>UML, OCL, MARTE, SysML, <strong>the Uncertainty Profile</strong></td>
</tr>
<tr>
<td>Model Based Product Line Testing</td>
<td>Feature Model, UML, the AspectSM Profile</td>
</tr>
</tbody>
</table>

Lessons Learnt:

- Poor capability of integrating methodologies and tools
- Difficult to think systematically since the beginning
  - Uncertainty taxonomy
- Always a challenge to evaluate a modeling methodology
  - Expressiveness, Usability, Applicability, Readability, etc.
- Which search algorithm to use in which situation
  - Large scale experiments to select algorithms for different situations
Acknowledgements

• Tao Yue, Shuai Wang, Bran Selic
• Man Zhang
• Former Colleagues: Lionel Briand, Andrea Arcuri, Hadi Hemmati, Zohaib Iqbal, Nina Holt
References

• **Modeling References**

• **Testing References**

• **Others**
Questions