Rafael Capilla

Universidad Rey Juan Carlos

Dynamic Variability Management Supporting Operational Modes of a Power Plant Product Line

Jan Bosch, Chalmers University of Technology, Sweden

Overview

- SPL Challenges
- Dynamic Variability
- TOSHIBA Product Line
- Dynamic Variability in Toshiba Power Plant
- Runtime Operational Modes
- Outcomes
- Summary & Research agenda
SPL Challenges

• Many of the today’s modern systems exhibit runtime behavior and adaptation capabilities

• Systems using context information: Mobile, Ubiquitous, Robots, WSN, Critical SoS in various application domains, Automotive, Mobile SW, Smart Cities, and many more…. 

• Self-adaptation + runtime behavior at post-deployment time is highly demanded…or just simply provide automatic updates of a new functionality 

• Conventional SPL using static variability models are not enough

10th VaMoS Workshop, Salvador de Bahia, Brasil, January 27-29, 2016
Dynamic Variability

*The big picture*

- **Dynamic SPL** is emerging as a valid paradigm to tackle runtime behavior of SPL products

- **Dynamic (runtime) variability** is the central technique of any DSPL

- **Context variability** combines conventional feature modeling with context features

- **Multiple binding times** are allowed and possible addressing different runtime concerns and system’s operational modes
Dynamic Variability

*What it can offer…*

- Changes in the structural variability at runtime (open variability models)
- Add/Remove/Change features dynamically beyond the simple activation/deactivation
- Handle collaborative features at real-time
- Enhanced feature dependencies (Static, Dynamic) and adding constraints at runtime
- Multiple binding times and the transition between them
**TOSHIBA Product Line**

*The Story of the EPG-SPL*

- Toshiba Electric Power Generation Product Line (EPG-SPL) started in 1977 (not like a modern SPL at that time)

- The EPG-SPL was included in the SPL HoF in 2008 (SEI)

- EPG-SPL was used for building control software for the development of new EPG systems
TOSHIBA Product Line

**EPG Requirements**

- Top-down and Bottom-up requirements belonging to power plant operators, device designers, etc

- Extensive interviews and surveys to elicit the requirements

- The design of the system and control solution produced a number of bottom-up requirements:
  - *Fault-tolerance and redundancy mgmt*
  - *Self-test initialization*
  - *Mgmt of EPG operational modes at runtime*
  - *Monitoring electric power and communication subsystems*
  - *Fuel handling and electric power transmission*
TOSHIBA Product Line

Extended EPG Requirements

• In 2013 more requirements were defined and grouped under 5 categories: system operation, system implementation, safety & dependability, execution environment and application software

• Elicit requirements using a top-down (TD) & bottom-up (BU) iterative process and propagated into the development phases

• We used a domain-specific language (DSL-EPG) to represent runtime requirements

• Dependencies between TD-BU requirements are difficult to maintain. DSL specifications were used to rewrite this part for new BU requirements
TOSHIBA Product Line

**EPG Operational Modes**

- The EPG-SPL encompasses different *operational modes* (e.g., sleeping, warm stop, wake-up, start-up, boiler ignition).

- Sequential logic between these operational modes

- DSL-EPG descriptions ➔ Executables codes (CHILD codes) while MOTHER codes are produced automatically

- A testing environment binds CHILD and MOTHER codes

- MOTHER codes include several MODEs to switch the execution of CHILD codes according to external conditions
Dynamic Variability in TOSHIBA Power Plant

- A runtime variability approach based on:
  - **Super-types** acting as classifiers of related features
  - Enable **add/remove/change features dynamically**
  - Support for **context features**
  - Support for **multiple binding times**
  - **Transition between binding** times at post-deployment time
Dynamic Variability in TOSHIBA Power Plant

Example

<table>
<thead>
<tr>
<th>Super-types</th>
<th>Basic types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>integer</td>
</tr>
<tr>
<td>Ambient</td>
<td>Integer, boolean</td>
</tr>
<tr>
<td>Multimedia</td>
<td>string</td>
</tr>
</tbody>
</table>

FOR all features in "ST: Ambient"
Configure (threshold values)
Activate_Feature (Comm_type.Event-based)
IF Check_Feature_States (All_features)=True
THEN Apply_changes (constraint_rule_set)

(a) Current WSN feature model

(b) WSN feature model modified after the inclusion of a new variant

10th VaMoS Workshop, Salvador de Bahia, Brasil, January 27-29, 2016
Dynamic Variability in TOSHIBA Power Plant

*Instantiation of the RunVar model*

- We mapped our runtime variability model to EPG-SPL runtime needs

<table>
<thead>
<tr>
<th>Runtime Variability model</th>
<th>Toshiba EPG-SPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changes in the structural variability</td>
<td>DSL-EPG descriptions and MODEs as super-types</td>
</tr>
<tr>
<td>Switch between binding times</td>
<td>MOTHER runtime framework</td>
</tr>
<tr>
<td>Runtime binding of variants</td>
<td>MOTHER codes bounded to values at runtime</td>
</tr>
<tr>
<td>Context features (not part of the model but derived from)</td>
<td>Runtime operational modes and events</td>
</tr>
</tbody>
</table>

*Adding features was done manually rewriting DSL descriptions*

*Now are partially automated adding/removing functionality in service*
Dynamic Variability in TOSHIBA Power Plant

Mapping Super-types to EPG-SPL

Top-down requirements

Meta-descriptions for generating MOTHER and CHILD codes

Super-Types

EPG MODEs

Heating boiler

Gas burner (op. Mode Start-up)

MOTHER switches between MODEs

Testing of the executed codes and analysis of the test results

Bottom-up requirements propagated to the next phase

Execution of the codes of the EPG using a simulator

Execution of CHILD

(1) Code generation
(2) Binding and configuration of the codes

10th VaMoS Workshop, Salvador de Bahia, Brasil, January 27-29, 2016
Dynamic Variability in TOSHIBA Power Plant

**Realize EPG runtime requirements**

- MOTHER switches between runtime MODEs, where variants are bounded at runtime

- MODEs control groups of CHILDren which implement different runtime requirements, such as:
  - Plant Master Status (PMS) monitor runtime functions such as emergency and daily start-stop
  - Alarm Group (ALG) control man-machine interactions
  - Operation Block (OB) as behavior performed by the sensory of the EPG
Runtime Operational Modes

- The variability of different system’s operational modes can be handled using multiple binding times.

- Our dynamic variability approach suggests different binding times at post-deployment time and the transition between them.
Runtime Operational Modes

**EPG-SPL operational modes at runtime**

- The **SCIA** (Status, Condition, Interaction and Control) is part of the mechanism of the DSL-EPG for handling the semantics of runtime needs using an event-oriented architecture.

**MAJOR ELEMENTS**

- **Status** = operational modes common to all plant operations (e.g., Start, Boiler ignition, Ignition preparation, etc.)

- **Conditions** = logical variables called Events

- **Events** = deviation of threshold values

- **Control** = logical expressions representing pre and post-conditions
Runtime Operational Modes

Operational semantics for runtime scenarios

Example of temporal relationship

DSL-EPG syntactic rules are satisfied by the SCIA

- status
  - trap
  - trigger
  - precondition
  - control

If the trap condition becomes true, the activity will be aborted

When the activity finishes successfully, the post-condition becomes true

If the timing, trigger and precondition are true, while trap condition is false, activity can start.

10th VaMoS Workshop, Salvador de Bahia, Brasil, January 27-29, 2016
Runtime Operational Modes

*Runtime scenarios*

- Runtime scenarios need a runtime environment to be executed
- MOTHER codes are used for handling the EPG-SPL runtime scenarios
- A repository includes runtime components for different EPG applications and type of power plants
- The EPG-SPL simulator is used to test runtime reconfigurations generated by MOTHER codes
Runtime Operational Modes

**EPG-SPL runtime workflow**

**MOTHER**
- **Meta-descriptions** defining the relationships between common framework and runtime components
- Common framework ((1)- fossil fuel app)
- Common framework ((n) - BWR type nuclear stations app)
- Common runtime support components

**CHILD**
- **Parameters** to customize a Meta-description for an app.
- **Code Generator**
  - Generated codes
  - DSL-EPG descriptions (runtime scenarios)
- **Workspace**
  - Merge codes from the code generator and the configuration control
- **Environment**
  - Testing
- **Binary codes**

**Control**
- **Configuration control** Combines app. framework and runtime components
- **Revision control**
- **Registration control**
- **Code inversion and analysis**
- **Testing**

10th VаMoS Workshop, Salvador de Bahia, Brasil, January 27-29, 2016
Runtime Operational Modes

**EPG-SPL environment supporting runtime scenarios**

---

10th VaMoS Workshop, Salvador de Bahia, Brasil, January 27-29, 2016
Outcomes

- We supported configuration and binding at runtime *(EPG-4)*
- Transitions between different binding times *(EPG-1, EPG-2)*
- Partial support of runtime changes in the structural variability adding new runtime features automatically using our Super-types model (open variability model) *(EPG-2)*
- Partial support for new operational modes that can be added automatically in a controlled manner *(EPG-2)*
Summary and Research Agenda

- Partial implementation of our runtime variability model at TOSHIBA EPG-SPL

- Difficulty to generate the runtime requirements but ability to accomodate EPG operational modes

- More effort is needed to support a full automation of changes in the variability model and define more categories of super-types

*It seems difficult and challenging to continue with this experience at TOSHIBA without the main researcher (Y. Matsumoto)*

10th VaMoS Workshop, Salvador de Bahia, Brasil, January 27-29, 2016
Summary and Research Agenda

RESEARCH AGENDA:

- Extend the implementation of the runtime variability approach to other domains

- Support automatic decision-making for multiple location of runtime variants in the feature model

- Provide more accurate mappings between application-specific operational modes and our dynamic variability model
In memorial of Yoshihiro Matsumoto

YOSHIHIRO

RAFA

KYO

JAN

Systems and Software Variability Management
Concepts, Tools and Experiences

Springer
Thank you

Q?