Extending Dynamic Software Product Lines with Temporal Constraints

Gustavo Sousa  Walter Rudametkin  Laurence Duchien

firstname.lastname@inria.fr

SEAMS 2017
(12th Int. Symp. on Soft. Eng. for Adaptive and Self-Managing Systems)
Adaptive Cloud Environments

• Cloud computing supports construction of customized adaptable environments

“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”[1]

• A cloud environment is a set of cloud services provisioned for running an application

Cloud Providers Configuration Variability

- Wide range of configurable cloud services
- Complex configuration rules and constraints
SPLs for Automated Cloud Configuration

C. Quinton et al. (2016) SALOON: a platform for selecting and configuring cloud environments.


J. García-Galán et al. (2016) Automated Configuration Support for Infrastructure Migration to the Cloud.
Dynamic Software Product Lines

• High variability with adaptive capabilities
Dynamic Software Product Lines

- **High variability with adaptive capabilities**

- **DSPL vs SPL**
  - Features can be (re)bound at runtime
  - Adaptive system vs systems family
  - Variability model central to both
Dynamic Software Product Lines

• **High variability with adaptive capabilities**

• **DSPL vs SPL**
  – Features can be (re)bound at runtime
  – Adaptive system vs systems family
  – Variability model central to both

• **Adaptation in DSPLs**
  – A context change is mapped to a request to include or exclude a set of features from the current configuration
  – SPL analysis is used to derive valid configurations
Cloud Computing Environment

• Reconfiguration mechanisms are provider-dependent and heterogeneous
  – May depend on initial or previous configurations
  – Alternative ways to reconfigure

• Compliance to variability model is not enough
  – Does not ensure valid and safe reconfigurations
Motivating Example

```
Heroku
  └── Application
      ├── Location
      │   ├── EU
      │   └── US
      ├── Framework
      │   ├── Java
      │   └── PHP
      └── Process Type
          ├── Kind
          │   ├── Web
          │   └── Worker
          └── Size
              ├── H1
              └── S1

Heroku Postgres
  └── ClearDB MySQL
      ├── H2
      └── S2
```

---

```
<table>
<thead>
<tr>
<th>Kind</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>H1</td>
</tr>
<tr>
<td>Worker</td>
<td>H2</td>
</tr>
<tr>
<td></td>
<td>S1</td>
</tr>
<tr>
<td></td>
<td>S2</td>
</tr>
</tbody>
</table>
```
Motivating Example

- Heroku
  - Application
    - Location
      - EU
      - US
    - Framework
      - Java
      - PHP
      - Ruby
    - Process Type
      - Kind
        - Web
        - Worker
      - Size
        - S1X
        - PM
        - S2X
    - Heroku Postgres
      - H1
      - H2
      - S1
      - S2
    - ClearDB MySQL
      - M1
      - M2
      - M3
      - M4
Motivating Example

Diagram showing:
- Heroku
- Application
- Process Type
  - Kind
    - Web
    - Worker
    - S1X
    - PM
    - S2X
  - Size
    - H1
    - H2
    - S2
    - S1
    - M1
    - M2
    - M3
    - M4
- Heroku Postgres
- ClearDB MySQL
Motivating Example

MySQL plan can only be upgraded

https://devcenter.heroku.com/articles/cleardb#upgrading-your-cleardb-database
Motivating Example

Heroku Postgres plan change
- PG Copy or Follower Changeover

https://devcenter.heroku.com/articles/upgrading-heroku-postgres-databases
Motivating Example

Changing the framework requires restarting the application

https://devcenter.heroku.com/articles/buildpacks#setting-a-buildpack-on-an-application
Motivating Example

Heroku

Application

Process Type

Kind

Web

Worker

S1X

PM

Size

S2X

H1

H2

S2

S1

M1

M2

M3

M4

Heroku Postgres

ClearDB MySQL

Framework

Java

PHP

Ruby

Location

EU

US
Limitations in DSPLs

• Seminal works on DSPLs highlight the need for validating transitions between system configurations
  – systems should evolve through safe migration paths\cite{6}
  – dynamic constraints on allowed transitions must be considered\cite{7}

• Validation is mostly limited to compliance to a variability model


Problem statement

• How to model constraints over the adaptation behavior?
  – Temporal dependencies between features and reconfiguration operations

• How to reason over a variability model with reconfiguration constraints to find reconfigurations that meet a given criteria?
  – e.g. reduced downtime or costs
Proposed approach

• Combine *variability models* with *temporal constraints* and *reconfiguration operations*
  – Leverage concepts and solutions from model checking
Feature Models and Transition Systems

- Feature model \( M = (F, C) \)
  - \( F \) is the set of features
  - \( C \subseteq \mathcal{P}(F) \)

![Feature Model Diagram]

- \( C_1 = \{A, E\} \)
- \( C_2 = \{A, E, F\} \)
- \( C_3 = \{A, B, C, E\} \)
- \( C_4 = \{A, B, C, E, F\} \)
- \( C_5 = \{A, B, D, E\} \)
- \( C_6 = \{A, B, D, E, F\} \)
DSPLs as Transition Systems

“A DSPL’s execution can be abstracted as a highly connected state machine where the states are the possible system configurations and the transitions the migration paths.”[6]

C1 = \{A, E\}
C2 = \{A, E, F\}
C3 = \{A, B, C, E\}
C4 = \{A, B, C, E, F\}
C5 = \{A, B, D, E\}
C6 = \{A, B, D, E, F\}

DSPLs as Transition Systems

C1 = \{A, E\}
C2 = \{A, E, F\}
C3 = \{A, B, C, E\}
C4 = \{A, B, C, E, F\}
C5 = \{A, B, D, E\}
C6 = \{A, B, D, E, F\}
Temporal properties

• A temporal property defines a condition over the executions of a transition system

  – Execution:

    $\rho = s_0 s_1 s_2 s_3 s_4 \ldots$

    $s_i \rightarrow s_{i+1}$ is a transition

  – A property is a set executions

  – A system exhibits a property if all its executions are part of the property set
Feature Models and Transition Systems

- Feature model \( M = (F, C) \)
- Transition system \( TS_M = (S, I, R, AP, L) \)

\[ S = I = C, \quad R = S \times S, \quad AP = F, \quad L(x) = x \]

C1 = \{A, E\}
C2 = \{A, E, F\}
C3 = \{A, B, C, E\}
C4 = \{A, B, C, E, F\}
C5 = \{A, B, D, E\}
C6 = \{A, B, D, E, F\}
Temporal properties

- A temporal property is a condition over the executions of a transition system

\[ P = \{ s_0s_1s_2s_3... \mid C \in L(s_i) \leftrightarrow D \notin L(s_{i+1}) \} \]
Temporal properties

- A temporal property is a condition over the executions of a transition system

\[
P = \{s_0s_1s_2s_3\ldots \mid C \in L(s_i) \iff D \not\in L(s_{i+1})\}\]

- \(C_1 = \{A, E\}\)
- \(C_2 = \{A, E, F\}\)
- \(C_3 = \{A, B, C, E\}\)
- \(C_4 = \{A, B, C, E, F\}\)
- \(C_5 = \{A, B, D, E\}\)
- \(C_6 = \{A, B, D, E, F\}\)
Linear Temporal Logic (LTL)

- Defines temporal properties over transition systems
- Combines propositional logic with temporal operators (always, eventually, until)

\[
\begin{align*}
\square A & \quad // \text{always } A \\
\square (M_2 \rightarrow \neg \bigcirc M_1) & \quad // \text{always (M2 is not followed by M1)} \\
\square (M_2 \rightarrow \neg \Diamond M_1) & \quad // \text{after } M_2, M_1 \text{ is not allowed}
\end{align*}
\]
DSPL with temporal properties

\[ \square (C \rightarrow \neg \bigcirc D) \]
\[ \square (D \rightarrow \neg \bigcirc C) \]
DSPL with temporal properties

\[ \square (C \rightarrow \neg \bigcirc D) \]
\[ \square (D \rightarrow \neg \bigcirc C) \]
Reconfiguration operations

- Doubly labeled transition systems\textsuperscript{[22]}

\[\text{C1} \rightarrow \text{C2} \rightarrow \text{C4} \rightarrow \text{C6} \]
\[\text{C3} \rightarrow \text{C2} \rightarrow \text{C4} \rightarrow \text{C5} \]

Reconfiguration operations

- Doubly labeled transition systems\textsuperscript{[22]}
  
  - $TS = (S, I, OP, R, AP, L)$
  
  - $OP$ is the set of reconfiguration operations in the DSPL
  
  - $R \subseteq S \times 2^{OP} \times S$

State/Event LTL

- SE-LTL can express temporal expressions over state and transition labels$^{[23]}$
  - Can combine reconfiguration operations and features in temporal constraints

Reconfiguration operations

\[ OP = \{ Activate F \} \]

\[ \square (C \rightarrow \neg \bigcirc D) \]
\[ \square (D \rightarrow \neg \bigcirc C) \]
\[ \square (\neg F \land \bigcirc F) \leftrightarrow Activate F \]
Examples

• Cannot downgrade MySQL plan
  – [](M2 -> !<>M1)
  – [](M3 -> !<>(M1 | M2))
  – [](M4 -> !<>(M1 | M2 | M3))
  – [](Change(ClearDBMySQL) -> UpgradeClearDB)

• Upgrade PostgreSQL
  – [](Change(HerokuPostgres) & (H1 | H2) -> PGCopy)
  – [](Change(HerokuPostgres) -> (PGCopy | FollowerChangeover))

• Change Location
  – [](Change(Location) -> MigrateApp)
Problem statement

• How to model constraints over the adaptation behavior?
  – Temporal dependencies between features and reconfiguration operations

• How to reason over a variability model with reconfiguration constraints to find reconfigurations that meet a given criteria?
  – e.g. reduced downtime or costs
Reasoning

- Reconfiguration request
  - Features to be included/excluded
Reasoning

• Reconfiguration request
  – Features to be included/excluded

• Cost-based constraints
  – Reconfiguration time, downtime, financial cost, etc
Reasoning

• Reconfiguration query: \( Q = (A, E, \phi) \)
  - \( A \): features to include
  - \( E \): features to exclude
  - \( \phi \): constraint over costs

• Example query: \( q = (\{C\}, \{D\}, \text{downtime} = 0) \)
Symbolic Representation

- Building the transition system for a feature model can be unfeasible
  - State-explosion problem

- Represent a transition system as a propositional formula
  - Use SAT solver to solve reconfiguration queries
Symbolic Representation

- Feature models\textsuperscript{[25]} and SE-LTL expressions\textsuperscript{[27]} can be represented as propositional formulas

\[ \tilde{M} \land \bar{x} \land \tilde{M}' \land \bar{s} \land \bar{q} \]

- \( \tilde{M} \) and \( \tilde{M}' \) represent the set of possible source and target states (configurations of the feature model \( M \))
- \( \bar{x} \) is the conjunction of LTL expressions
- \( \bar{s} \) represents the current state
- \( \bar{q} \) represents the reconfiguration query (pseudo-boolean encoding)


Problem statement

• How to model constraints over the adaptation behavior?
  – Temporal dependencies between features and reconfiguration operations

• How to reason over a variability model with reconfiguration constraints to find reconfigurations that meet a given criteria?
  – e.g. reduced downtime or costs
Evaluation
Evaluation

• Case study on Heroku PaaS
  – feasibility for modeling reconfiguration constraints
  – performance of reasoning
Evaluation

• Case study on Heroku PaaS

• Feature Model extracted from documentation
  – 7 available regions, 11 programming frameworks, 6 container sizes
  – reconfiguration constraints
  – 161 addon services (data storage, networking, security, ...)
  – 1036 features, 134 cross-tree constraints, 124 temporal constraints
Evaluation

• Case study on Heroku PaaS

• Feature Model extracted from documentation

• Simulate context changes
  – 4 adaptation scenarios
  – 5 reconfiguration queries
  – 3 utilization profiles
  – 12 executions
Evaluation

• Case study on Heroku PaaS

• Feature Model extracted from documentation

• Simulate context changes

• Adaptation scenarios
  – Change in database size requires new database plan
  – Request for a new feature not available in current region
  – Change in programming framework and database
  – Scaling up application container
Evaluation

- Case study on Heroku PaaS
- Feature Model extracted from documentation
- Simulate context changes
- Adaptation scenarios
- Reconfiguration queries
  - No constraints
  - Constraints over price
  - Constraints over downtime/price
  - Optimize on price
  - Optimize on downtime/price
Evaluation

- Case study on Heroku PaaS
- Feature Model extracted from documentation
- Simulate context changes
- Adaptation scenarios
- Reconfiguration queries
- Application utilization profiles
  - Database size, application size, startup time, etc...
Evaluation

• Case study on Heroku PaaS

• Feature Model extracted from documentation

• Simulate context changes

• Adaptation scenarios

• Reconfiguration queries

• Application utilization profiles
  – DBSize: 10GB, AppSize: 100 MB, AppStartUp: 15
  – DBSize: 100GB, AppSize: 200 MB, AppStartUp: 30s
  – DBSize: 2TB, AppSize: 500 MB, AppStartUp: 60s
Results

<table>
<thead>
<tr>
<th>Process step</th>
<th>Avg</th>
<th>StdDev</th>
<th>Min</th>
<th>Max</th>
<th>#Exec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build Trans System</td>
<td>8777.31</td>
<td>303.71</td>
<td>8262</td>
<td>10308</td>
<td>720</td>
</tr>
<tr>
<td>- Process FM</td>
<td>244.75</td>
<td>28.57</td>
<td>191</td>
<td>552</td>
<td></td>
</tr>
<tr>
<td>- Process LTL</td>
<td>8533.57</td>
<td>291.81</td>
<td>8025</td>
<td>10023</td>
<td></td>
</tr>
<tr>
<td>All Queries</td>
<td>183.34</td>
<td>50.20</td>
<td>118</td>
<td>389</td>
<td>720</td>
</tr>
<tr>
<td>- Build</td>
<td>83.05</td>
<td>50.69</td>
<td>27</td>
<td>227</td>
<td></td>
</tr>
<tr>
<td>- Solve</td>
<td>100.29</td>
<td>37.13</td>
<td>5</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>Q wo/ Constraints</td>
<td>140.97</td>
<td>12.93</td>
<td>118</td>
<td>198</td>
<td>144</td>
</tr>
<tr>
<td>- Build</td>
<td>32.73</td>
<td>4.56</td>
<td>27</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>- Solve</td>
<td>108.24</td>
<td>11.58</td>
<td>90</td>
<td>153</td>
<td></td>
</tr>
<tr>
<td>Q w/ Constraints</td>
<td>224.23</td>
<td>55.55</td>
<td>128</td>
<td>389</td>
<td>288</td>
</tr>
<tr>
<td>- Build</td>
<td>140.41</td>
<td>27.65</td>
<td>80</td>
<td>227</td>
<td></td>
</tr>
<tr>
<td>- Solve</td>
<td>83.82</td>
<td>53.13</td>
<td>5</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>Q w/ Optimization</td>
<td>163.63</td>
<td>13.26</td>
<td>136</td>
<td>230</td>
<td>288</td>
</tr>
<tr>
<td>- Build</td>
<td>50.85</td>
<td>7.11</td>
<td>38</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>- Solve</td>
<td>112.77</td>
<td>10.21</td>
<td>94</td>
<td>166</td>
<td></td>
</tr>
</tbody>
</table>
# Results

<table>
<thead>
<tr>
<th>Process step</th>
<th>Avg</th>
<th>StdDev</th>
<th>Min</th>
<th>Max</th>
<th>#Exec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build Trans System</td>
<td>8777.31</td>
<td>303.71</td>
<td>8262</td>
<td>10308</td>
<td>720</td>
</tr>
<tr>
<td>- Process FM</td>
<td>244.75</td>
<td>28.57</td>
<td>191</td>
<td>552</td>
<td></td>
</tr>
<tr>
<td>- Process LTL</td>
<td>8533.57</td>
<td>291.81</td>
<td>8025</td>
<td>10023</td>
<td></td>
</tr>
<tr>
<td>All Queries</td>
<td>183.34</td>
<td>50.20</td>
<td>118</td>
<td>389</td>
<td>720</td>
</tr>
<tr>
<td>- Build</td>
<td>83.05</td>
<td>50.69</td>
<td>27</td>
<td>227</td>
<td></td>
</tr>
<tr>
<td>- Solve</td>
<td>100.29</td>
<td>37.13</td>
<td>5</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>Q wo/ Constraints</td>
<td>140.97</td>
<td>12.93</td>
<td>118</td>
<td>198</td>
<td>144</td>
</tr>
<tr>
<td>- Build</td>
<td>32.73</td>
<td>4.56</td>
<td>27</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>- Solve</td>
<td>108.24</td>
<td>11.58</td>
<td>90</td>
<td>153</td>
<td></td>
</tr>
<tr>
<td>Q w/ Constraints</td>
<td>224.23</td>
<td>55.55</td>
<td>128</td>
<td>389</td>
<td>288</td>
</tr>
<tr>
<td>- Build</td>
<td>140.41</td>
<td>27.65</td>
<td>80</td>
<td>227</td>
<td></td>
</tr>
<tr>
<td>- Solve</td>
<td>83.82</td>
<td>53.13</td>
<td>5</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>Q w/ Optimization</td>
<td>163.63</td>
<td>13.26</td>
<td>136</td>
<td>230</td>
<td>288</td>
</tr>
<tr>
<td>- Build</td>
<td>50.85</td>
<td>7.11</td>
<td>38</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>- Solve</td>
<td>112.77</td>
<td>10.21</td>
<td>94</td>
<td>166</td>
<td></td>
</tr>
</tbody>
</table>
### Results

<table>
<thead>
<tr>
<th>Process step</th>
<th>Avg</th>
<th>StdDev</th>
<th>Min</th>
<th>Max</th>
<th>#Exec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build Trans System</td>
<td>8777.31</td>
<td>303.71</td>
<td>8262</td>
<td>10308</td>
<td>720</td>
</tr>
<tr>
<td>- Process FM</td>
<td>244.75</td>
<td>28.57</td>
<td>191</td>
<td>552</td>
<td></td>
</tr>
<tr>
<td>- Process LTL</td>
<td>8533.57</td>
<td>291.81</td>
<td>8025</td>
<td>10023</td>
<td></td>
</tr>
<tr>
<td>All Queries</td>
<td>183.34</td>
<td>50.20</td>
<td>118</td>
<td>389</td>
<td>720</td>
</tr>
<tr>
<td>- Build</td>
<td>83.05</td>
<td>50.69</td>
<td>27</td>
<td>227</td>
<td></td>
</tr>
<tr>
<td>- Solve</td>
<td>100.29</td>
<td>37.13</td>
<td>5</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>Q wo/ Constraints</td>
<td>140.97</td>
<td>12.93</td>
<td>118</td>
<td>198</td>
<td>144</td>
</tr>
<tr>
<td>- Build</td>
<td>32.73</td>
<td>4.56</td>
<td>27</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>- Solve</td>
<td>108.24</td>
<td>11.58</td>
<td>90</td>
<td>153</td>
<td></td>
</tr>
<tr>
<td>Q w/ Constraints</td>
<td>224.23</td>
<td>55.55</td>
<td>128</td>
<td>389</td>
<td>288</td>
</tr>
<tr>
<td>- Build</td>
<td>140.41</td>
<td>27.65</td>
<td>80</td>
<td>227</td>
<td></td>
</tr>
<tr>
<td>- Solve</td>
<td>83.82</td>
<td>53.13</td>
<td>5</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>Q w/ Optimization</td>
<td>163.63</td>
<td>13.26</td>
<td>136</td>
<td>230</td>
<td>288</td>
</tr>
<tr>
<td>- Build</td>
<td>50.85</td>
<td>7.11</td>
<td>38</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>- Solve</td>
<td>112.77</td>
<td>10.21</td>
<td>94</td>
<td>166</td>
<td></td>
</tr>
</tbody>
</table>
Results

• Temporal constraints enhance modeling of DSPLs
  – Compact notation for constraints over transitions
  – Support for reasoning over reconfiguration operations

• Performance is acceptable in the cloud context
  – Implementation can be improved

• Threats to validity
  – Case study is not exhaustive and considers only cloud computing
Conclusion & Perspectives

• Temporal constraints in DSPL
  – Better modeling of adaptive behavior
  – Reasoning over adaptation alternatives
Conclusion & Perspectives

• Temporal constraints in DSPL
  – Better modeling of adaptive behavior
  – Reasoning over adaptation alternatives

• Cardinality-based feature models

• Multi-cloud environment adaptation
Questions

Gustavo Sousa
gustavo.sousa@inria.fr

More information
http://researchers.lille.inria.fr/sousa/seams2017/