A Framework to Support the Development and Evolution of Self-adaptive Data-intensive Systems

Marco Mori and Anthony Cleve
PReCISE Research Center, University of Namur
{mmo,acl} @info.fundp.ac.be
Introduction

- Nowadays software systems are:
  - Adaptive (context-aware, ubiquitous)
  - Data-intensive (big-data, cloud computing)
- Self-adaptive systems
  - Design time: different software alternatives to satisfy user requirements in different contexts
  - Run-time: selection of the most suitable alternative based on the current context
- Different software alternatives possibly need distinct/partial portions of a global
Introduction

• Context-aware database
  • Supporting the adaptivity over *data* based on context, user tasks and user preferences

• Problem:
  • A big number of software alternatives
  • Large database

• Software and data adaptation

• Link variability of *requirements* to variability of *data*
Motivating scenario’s

- Large amount of data (data variability) and large number of software alternatives (requirements variability)
- Context-dependent access to data:
  - Device characteristics
  - Physical environment
  - User role
  - User task
- Two possible case studies:
  - eHealth: user roles (Doctor, Nurse, Patient, Secretary,...), user tasks (Visits, Surgery, Check-up, Administration activities,...)
  - SmartCity: user roles (Citizen, Tourist, Employer,...), user tasks (Transport schedules consulting, Restaurants consulting, Tourist tour,...)
Framework basics

- Context
  - User role and task
  - Device characteristics
  - Physical environment

- Software functionalities
  - Feature engineering perspective

- Sub-set of the database
  - Sufficient (w.r.t. context)
  - Consistent (w.r.t. the large database)
Filtering design phase

- User requirements
- Data Modeling
- Feature Modelling
- Feature To Data Mapping
- Context Modeling
- Context To Feature Mapping
Data modelling

[Diagram showing the processes of conceptual analysis, logical design, physical design, and coding, with a conceptual schema, a logical schema, and a physical schema.]
Data modelling
Feature modelling

- A feature is the smallest unit of behaviour as it can be perceived by a user, i.e., \( f=(R,P,V) \):
  - \( R \) is a functional, non-functional or a specific quality requirement
  - \( P \) is the presence condition expressed as a contextual constraint requirement
  - \( V \) is the excerpt of data of the conceptual schema required by the feature
Feature model

- eHealth Patient Management
  - Visit Info
    - Text Case History
      - HqMRI
      - LqMRI
    - MR Images
      - HQMRI
      - LqMRI
    - Rad Images
      - HQRad
      - LqRad
  - Case History
  - Therapy Management
  - Administrative Info
    - Basic A
    - Complete A
  - Dep Management
    - Cardiology
    - Orthopedy
  - Diet Management
# Feature to data mapping

<table>
<thead>
<tr>
<th>Feature</th>
<th>Presence condition</th>
<th>Focus Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>VisitInfo</td>
<td>Task=Check-up</td>
<td>{Visit, Doctor, Specialization}</td>
</tr>
<tr>
<td>CompleteA</td>
<td>Role=Secretary OR Task=PersInfoCons</td>
<td>{BedAssignment, Patient}</td>
</tr>
<tr>
<td>BasicA</td>
<td>Task=Emergency OR Task=Check-up OR Location=In</td>
<td>{Patient, SSN, Patient.Name, Patient.Age}</td>
</tr>
<tr>
<td>TextCaseHistory</td>
<td>Task=Emergency OR Task=Check-up OR Location=Out</td>
<td>{CaseHistory}</td>
</tr>
<tr>
<td>HqMRI</td>
<td>(Role=Radiologist AND Location=MRI Room) OR (Task=Check-up AND Device=Desktop AND Location!=Out)</td>
<td>{MriHighQ, MriImm}</td>
</tr>
<tr>
<td>LqMRI</td>
<td>(Role=Radiologist AND Location=MRI Room) OR (Task=Check-up AND Device=Mobile AND Location!=Out)</td>
<td>{MriLowQ, MriImm}</td>
</tr>
<tr>
<td>HqRad</td>
<td>(Role=Radiologist AND Location=Rad Room) OR (Task=Check-up AND Device=Desktop AND Location!=Out)</td>
<td>{RadHighQ, RadImm}</td>
</tr>
<tr>
<td>LqRad</td>
<td>(Role=Radiologist AND Location=Rad Room) OR (Task=Check-up AND Device=Mobile AND Location!=Out)</td>
<td>{RadLowQ, RadImm}</td>
</tr>
<tr>
<td>Therapy Management</td>
<td>Role=Doctor OR Role=Nurse</td>
<td>{MedApp, Diagnosys, Therapy, LongStay, DayHospital}</td>
</tr>
<tr>
<td>Cardiology</td>
<td>Dept=Cardiology</td>
<td>{Department, Cardiology}</td>
</tr>
<tr>
<td>Orthopedy</td>
<td>Dept=Orthopedy</td>
<td>{Department, Orthopedy}</td>
</tr>
<tr>
<td>DietManagement</td>
<td>Role=Doctor OR Role=Nurse OR (Role=Patient AND Location=Out)</td>
<td>{DietPlan, DayPlan, MealPlan}</td>
</tr>
</tbody>
</table>

![Diagram](image-url)
Feature configurations
Context to feature mapping

- The context describes user role, user task, device characteristics, location, etc...
- Each context state is an element in $C = D_1 \times D_2 \times \ldots \times D_n$, where $D_1, D_2, \ldots, D_n$ are the context dimensions.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Presence condition</th>
<th>Focus Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>VisitInfo</td>
<td>Task=Check-up</td>
<td>(Visit, Doctor, specialisation)</td>
</tr>
<tr>
<td>CompleteA</td>
<td>Role=Secretary OR Task=PersInfoCons</td>
<td>(BedAssignment, Patient)</td>
</tr>
<tr>
<td>BasicA</td>
<td>Task=Emergency OR Task=Check-up</td>
<td>(Patient, SSN, Patient Name, Patient Age)</td>
</tr>
<tr>
<td>TextCaseHistory</td>
<td>Task=Emergency OR Task=Check-up OR Location=Out</td>
<td>(CaseHistory)</td>
</tr>
<tr>
<td>HqMRI</td>
<td>(Role=Radiologist AND Location=MRI Room) OR (Task=Check-up AND Device=Desktop AND Location!=Out)</td>
<td>(MriHighQ, Mrlmm)</td>
</tr>
<tr>
<td>LqMRI</td>
<td>(Role=Radiologist AND Location=MRI Room) OR (Task=Check-up AND Device=Mobile AND Location!=Out)</td>
<td>(MrllowQ, Mrlmm)</td>
</tr>
<tr>
<td>HqRad</td>
<td>(Role=Radiologist AND Location=Rad Room) OR (Task=Check-up AND Device=Desktop AND Location!=Out)</td>
<td>(RadHighQ, Radlimm)</td>
</tr>
<tr>
<td>LqRad</td>
<td>(Role=Radiologist AND Location=Rad Room) OR (Task=Check-up AND Device=Mobile AND Location!=Out)</td>
<td>(RadLowQ, Radlimm)</td>
</tr>
<tr>
<td>Therapy Management</td>
<td>Role=Doctor OR Role=Nurse</td>
<td>(MedApp, Diagnosys, Therapy, LongStay, DayHospital)</td>
</tr>
<tr>
<td>Cardiology</td>
<td>Dep=Cardiology</td>
<td>(Department, Cardiology)</td>
</tr>
<tr>
<td>Orthopedy</td>
<td>Dep=Orthopedy</td>
<td>(Department, Orthopedy)</td>
</tr>
<tr>
<td>DietManagement</td>
<td>Role=Doctor OR Role=Nurse OR (Role=Patient AND Location=Out)</td>
<td>(DietPlan, DayPlan, MealPlan)</td>
</tr>
</tbody>
</table>
Context-Features-Data

Ci

Cj

Diagram showing the relationship between context, features, and data with various nodes and connections.
Context-Features-Data

Ci

[Diagram with context, features, and data elements]

[Network diagrams with various nodes and connections]
Filtering process
Decision-making process

- What: Selecting the *most suitable* configuration of features (data) with the aim of improving the overall stability of the reconfiguration process
- Idea: anticipating future variations of context and the corresponding required features (data)

- Exploiting a *probabilistic task model* where:
  - each state is a different task annotated with the minimum set of required features
  - transitions show the probability of task change
Decision-making process

- Input: admissible configurations of features according to:
  - Device space limit
  - Minimum set of required features at the current state
- How: Multi-objective optimization technique that considers for each admissible configuration:
  - its data-requirement distance w.r.t. current deployed configuration
  - its data-requirement distance w.r.t. the (probable) future states
- Upon a task variation, we select the configuration with the highest fitness
- Emerging results show under which conditions (e.g., shared data required by tasks) it is valuable to consider both current and probable future tasks to improve the stability
On-going work

• What?
  • (a) Supporting database adaptation
  • (b) Improving performance of database adaptation
  • (c) Supporting program comprehension and bug detection

• How?
  • (a) Implementing the filtering algorithms (DBMain[1], MySql, CSP[2])
  • (b) Determining the probabilistic task model from the data-manipulation behavioural traces (SQL-statement traces, FCA[3])
  • (c) Determining the process model through process mining techniques from the data-manipulation behavioural traces (SQL-statement traces, ProM[4], FCA)

Summary

- We link together context, features and data
- We support the designing of self-adaptive data intensive applications
- We support context-driven database adaptation
  - Decision-making process to choose the best reconfiguration
  - Algorithms to create the subset of the large database
  - Consistency of the subset of data
- We support the reification of the data-manipulation behavior:
  - Migrating non self-adaptive data-intensive applications