How to Capture Context and Context-dependent Behavior

Tetsuo Tamai (Hosei University)
Kumiki Project Series

- Grant-in-Aid for Scientific Research (KAKEN)
  - Sponsored by MEXT (Ministry of Education, Culture, Sports, Sciences and Technology)
- Kumiki 1 (2001/10-2006/3)
- Kumiki 2 (2006/4-2010/3)
- Kumiki 3 (2010/4-2014/3)
Kumiki (組木)
“High Reliability Component-Based Software Engineering”

Members:
T. Tamai*, E. Shibayama, H. Masuhara, S. Nakajima, N. Ubayashi, A. Igarashi
Major Objectives

- How to design components and composition
- How to enhance reliability of components and systems integrating components
Approaches

- Clean and flexible modularization to conquer structural complexity
- Formal reasoning to conquer behavioral complexity
Major Results

- **Component & composition design**
  - Collaboration Model: Epsilon (T. Tamai et al.)
  - Aspect Oriented Model (H. Masuhara et al.)

- **Formal verification**
  - Model checking component framework (Nakajima & Tamai)
  - Variant parametric type system (A. Igarashi)
"Aspect-Oriented Software Development for Productivity and Reliability"

Members:
T. Tamai*, S. Chiba, H. Masuhara, S. Nakajima, N. Ubayashi, A. Igarashi
Motivation

- Software is in everywhere but getting invisible ever more.
- Much pressure on software providers w.r.t. shorter development cycle, variety of versions and high reliability
Our Approach

- **Clear modularization**
  - modules corresponding to features and concerns
  - clear boundary
  - correct internal structure and behavior
  - flexible composition
  - method for assuring safety and reliability of systems as composites
Targets & Results

- Propose new design methods, languages and verification methods for AOP
- Develop aspects for security, redundancy and other measures for reliability
- Integrate methods and tools to establish AOSD methodology covering requirements to evolution
Kumiki 3

“Modularization integrating hierarchical and crosscutting concerns in the post-aspect-orientation era”

Members:
S. Chiba*, H. Masuhara, T. Tamai,
S. Nakajima, N. Ubayashi, A. Igarashi,
T. Kamina
Target

- New language mechanism integrating hierarchical and cross-cutting modularization
  - Predicate dispatching mechanism
  - Context Oriented Programming
- Supporting theory
- Modeling method
AOP has been producing practical results but not so widely accepted in industry as OOP.

The reason may be complexity brought by two different modularization mechanisms for cross-cutting and hierarchical concerns.
Work divisions

- **Modeling**
  - Tamai, Nakajima

- **Design & Implementation**
  - Chiba, Masuhara, Ubayashi

- **Theory**
  - Igarashi, Kamina

- **International collaborator:** Robert Hirschfeld
Results

- **Modularization mechanism**
  - Language GluonJ extending predicate dispatching
  - Type theory for GluonJ

- **Extending Context-Oriented Programming**
  - Language EventCJ integrating COP and event-driven mechanism
  - Fundamental language model L, integrating method dispatching by class and by layer (with R. Hirschfeld)

- **UML4COP**: modeling method for COP

- **CJAdviser**: debugging tool for COP
What is Context?

- Context diagram in “Structured Analysis” and “Problem Frame”
  - relation between the system to be developed and the outer world where the system be deployed.
Context Diagram of Problem Frame

by Fred the Oyster
Recent Focus on “Context Change”

- context change as
  - driving force for behavior adaptation
- motivation
  - mobile computing
  - ubiquitous (pervasive) computing
- context-awareness is required to adaptive systems
Example Problems

- [Kamina et al. 2015]
  ◆ Conference guide system
  ◆ CJEdit program editor
  ◆ Maze-solving robot simulator

- [Sutcliffe et al. 2006]
  ◆ e-Mail for cognitively disabled users
  ◆ Navigation support system

- [Tamai & Monpratarnchnai 2014]
  ◆ Traffic jam monitoring
What Determines Context?

- **Location**
  - GPS, indoor/outdoor, floor/room/section, home/office, ...

- **Time**
  - season, day/night, weekday/weekend, ...

- **Natural environment**
  - temperature, weather, ...

- **Technical environment**
  - online/local, device type, battery status, ...

- **Social environment/personal properties**
How Does it Affect Behavior?

- **In COP,**
  - layers to modularize context-dependent behavior
  - layers dynamically activated/deactivated
Interactive Context

- Context determined by actors within and their interaction
  - fit to social environment but also others e.g. location and time context of home/office
  - behavior of actors determined by their roles
    - wife & husband, employer & employee
Context and Roles

Epsilon Model

- **Contexts** encapsulate collaboration fields enclosing a set of **roles**.
- **Objects** freely enter a context by **binding** to roles and leave a context by **unbinding** the ties to roles.
- **Objects** can belong to multiple contexts at a time.
- **Contexts** (with roles) are independent reuse components to be deployed separately from objects.
Context, Roles and Binding

role:manager

role:contractor

context

binding

object

multiple binding
How Context Changes

- By entering one context instance
- Switching from one context to another
- Context attribute values change
How Behavior Changes

By entering a context and binding with a role, an object obtains new behavior of the role or changes its behavior by the overriding mechanism.
Example for Comparison

“Has-network” context of the conference guide system

♦ COP
  - HasNetwork layer is defined and condition for activation declared
  - A class declares the layer

♦ Epsilon
  - Network context is defined with node roles
  - An object binds with a node role instance