Leveraging Activity Theory and Cloud Computing for Supporting Distributed Software Development Teams

Muhammad Ali Babar
IT University of Copenhagen, Denmark
Background Brief

M. Ali Babar

Associate Professor @ ITU

Work engagements:

ITU, CPH: Dec. 2009 …
NICTA, Australia: 2003 – 2007
JRCASE, Macquarie University: 2001 – 2003
Various industrial roles in IT: Prior to 2001

Research in software architecture, Service Orientation, Cloud Computing, and Software Development Paradigm

http://malibabar.wordpress.com
Outline

- Area of Research and Motivators
- Activity Theory and Activity Based Computing (ABC)
- An Infrastructure for Supporting ABC for GSE
- Leveraging Cloud Computing for GSE
- Road Ahead and Potential Challenges
Tools for Distributed Architecting Process

Main activities of software architecture process:
- Architectural Analysis
- Architectural Synthesis
- Architectural Evaluation
- Architectural Realization
- Architectural Evolution

Types of tools required:
- Architectural knowledge repositories
- Decision support systems
- Architecture modeling tools
- Web 3.0 technologies
Tools for Knowledge Ecosystem

Private Ecosystem A
- Company
- Employee
- Implementing
- create customized AK input form
- share AK

Public Ecosystem
- Collaboration
- Private Ecosystem B
- Private Ecosystem C
- View AK
- Modeling

IDE
- AK Consume
- AK Extraction

Modeling Tool
- AK Consume
- AK Extraction

Requirement
- Integration

CM/Issue Tracking
<table>
<thead>
<tr>
<th>Awareness</th>
<th>Project Management</th>
<th>Communication</th>
<th>Version Control System</th>
<th>Process Management</th>
<th>Remote Access</th>
<th>Risk Analysis</th>
</tr>
</thead>
</table>
Research Focus

- Research Motivation
  - How can the limitations of desktop centric tools be addressed for designing next generation of GSE tools?
  - How to provide tools more effectively and efficiently?
- Potential areas of exploration
  - Application of Activity Theory
  - Leveraging Cloud Computing
Activity Theory

Vygotsky

Instrument

Object

Leontiev [1978]

Instrument

Subject

Community

Division of Labor

Object

Rules

Engeström [1999]

Activity

Motive

Goal/Purpose

Operation

Condition

Action/Process
Activity-Centered Resource Aggregation

Activity Suspension and Resumption

Activity Sharing

Activity Roaming

Activity Awareness
Application of Activity Theory

Diagram A: Software architecture design framework
- Stakeholders, organization staff
- Capabilities: experience, and competences

Diagram B: Analysis
- A. Design
- A. Synthesis
- A. Evaluation
- Plan AE
- Prepare and manage results

Diagram C: Software architecture evaluation framework
- Evaluation manager
- Candidate architecture
- Validated architecture
- Context rules, e.g., organizational, ethical, ...
- Stakeholders, project manager, evaluation team, ...
- Capabilities: experience, and competences

Diagram D: Activities
- A. Design
- Activity
- Plan AE
- A. Evaluation
- Action
- Interpret artifacts
- Prepare reports
- Present results
- Prepare and manage results
Application of Activity Theory

Design → Activity → A. Evaluation → Plan AE → Action → Operation

Prepare and manage results → Interpret artifacts → Prepare reports → Present results

Software architecture evaluation framework

Evaluation manager → Candidate architecture → Validated architecture

Context rules, e.g., organizational

Stakeholders, project manager, evaluation

Capabilities, experience, and
Identifying Requirements

System quality requirements
- Unity
- Flexibility
- Scalability
- Intuitivity

System requirements for computer-mediated teamwork
- Collaboration
- Coordination
- Communication
## Data Model and Activity System

<table>
<thead>
<tr>
<th>Activity system</th>
<th>Data model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>User</td>
</tr>
<tr>
<td>Objective</td>
<td>Ecology</td>
</tr>
<tr>
<td>Outcome</td>
<td>Activity description</td>
</tr>
<tr>
<td>Control</td>
<td>Application</td>
</tr>
<tr>
<td>Community</td>
<td>Users part of the Ecology</td>
</tr>
<tr>
<td>Roles</td>
<td>-</td>
</tr>
<tr>
<td>Division of labor</td>
<td>Relationship</td>
</tr>
</tbody>
</table>
Leveraging Cloud Computing for GSE

IT UNIVERSITY OF COPENHAGEN
Why Cloud Computing Matters for GSE:

- Cloud computing has huge potential for GSE
- Testing as a Service (TaaS)
  - Stress testing with cloud infrastructure
- Enabling offshore testing of application for sensitive data
- Collaborative environment in Clouds
  - Just-in-Time (JIT) tool composition
- Processes, tools & context aligned

Source: http://aws.amazon.com/solutions/case-studies/8kmiles/
NexGen Infrastructure for GSE

- Context
  - Providing supportive technologies to GSE teams

- Challenges
  - Dozens of different tools used without easy integration
  - Misalignment between tools, processes, and culture
  - No Just-In-Time (JIT) composition as a service

- Proposed solution
  - Cloud-based infrastructure for Tools as a Service (TaaS)
On demand tools acquisition & access to a wide range of tools

Alignment among processes and tools for each project
Artefacts traceability across multiple sites and multiple organizations

Implicit support for real-time awareness and collaboration
Access to sensitive and massive amount of data without violating the privacy or data movement

Easy access to expensive and sensitive technological innovation

Support for inter-Organizational knowledge ecosystem
Requirements & Potential Solutions 1/2

- Support for multiple organizations & teams
  - Multi-tenancy for providing isolation between multiple services
- Privacy and security handling services
- Tools versioning and bundling
  - Version management for maintaining partitioning between different versions of tools & combining them as a tool suite
- Integration with commercially available tools
  - Platform neutral APIs and compatible data structures
- Tools working with private data and artefacts
  - Workflow like distributable data processing services
Support for multiple types of persistence methods
  - Design & exploit a multi-tenancy database driver

Accessibility from multiple types of devices
  - Dynamic distribution of processing load on devices & clouds

Hooks for implementing or inter-changing with services for defining & selecting optimal configuration strategy for tools

Transform responses to formats recognizable by clients

Compliance with service level agreement (SLA) – QoS
  - Specify machine readable & dynamically changeable SLAs

Continuous monitoring & dynamic execution of service according to SLA specification (e.g., scalability rules / elasticity algorithms)
An Overview of a Reference Architecture

Persistence Handling Services
SLA Compliance & Brokerage Services
Data Management Services
APIs For Integration With External Tools
Monitoring Services
SLA Modeling & Specification Services

Multi-tenancy Management
Versioning Management
Services Management Engine
Services Composition Modules

Virtual Nodes Manager
Response Transformation for Mobile Devices

Host Type 3
MS Project
IBM Rational Suite

Expensive/Sensitive Hardware
Project Managers’ Tablets
Developers’ Terminals
An Overview of a Reference Architecture

Persistence Handling Services
SLA Compliance & Brokerage Services
Multi-tenancy Management
Versioning Management
Virtual Nodes Manager
Response Transformation for Mobile Devices
Data Management Services
Services Management Engine
Services Composition Modules
APIs For Integration With External Tools
Monitoring Services
SLA Modeling & Specification Services
An Overview of a Reference Architecture

Virtual Tools and Platforms Hosted on Cloud

Persistence Handling Services
- Multi-tenancy Management
- Virtual Nodes Manager
- Response Transformation for Mobile Devices

SLA Compliance & Brokerage Services
- Versioning Management

Data Management Services
- Services Management Engine
- Services Composition Modules

APIs For Integration With External Tools

Monitoring Services
- SLA Modeling & Specification Services

Middleware Infrastructure

Expensive/Sensitive Hardware

Private Cloud Hosting
Sensitive Data

Developers' Terminals

Host Type 1
- Tomcat
- Eclipse
- MySQL

Host Type 2
- Visual Studio
- IIS
- VSS
- SQL Server

Host Type 3
- MS Project
- IBM Rational Suite

Project Managers' Tablets

Developers' Terminals
Road Ahead
More solid theoretical foundation of applying ABC for building SE tools in general and GSE tools in particular

Further refinement of architectural design and features, e.g. cloud-enabled infrastructure

Empirical evidence for viability and potential benefits
TaaS aren’t SaaS

- TaaS contains executable artefacts – what kinds of security mechanisms are required?
- TaaS needs integration with other tools and middleware components that may belong to multiple organizations
- TaaS generated artefacts can require unpredictable resources
- TaaS needs different kinds of SLAs
- TaaS needs integration with large number of proprietary and Open Source tools – licensing provisions and fees
Tools & Process Support for Migration

SaaS Quality Requirements

Existing Architecture

- Generates and posts updates to external clients like Twitter.
- Provides visualization of different metrics through Gollum.
- Provides data abstraction at a high level.
- Provides data abstraction at a low level.
- Manages persistence and handles operations associated with user data (e.g., account creation, user authentication, saving and retrieving data sent by sensors).

Target Platform(s)

- Patterns for achieving quality requirements in SaaS applications.
- Patterns for achieving self-autonomous features in SaaS applications (internal management).
- APIs for internal management of applications to link them with platform management for satisfying SLAs of tenants of platform as well as tenants of individual applications.

Forward Engineering Engine

Modified Architecture

Patterns Catalogue
Acknowledgements

- ABC4GSD work is being performed with Paolo Tell
- TaaS work is being performed with Aufeef Chauhan
- Several academic and industrial colleagues have provided useful inputs and feedbacks
Questions

M. Ali Babar

alibabar.m@gmail.com
malibabar.wordpress.com
Supporting Multi-Tenancy

Handling & serving requests from a single server

Parsing & validating requests from services

Access control & authentication rules

Service management

Multi-tenant DB drivers & Indexes
Supporting Multi Types of Devices

- Platform neutral interface
- Context monitoring & configuration
- Resource optimization
- Self management client components
- Responses composition