Technical Integration of Engineering Environments Across Engineering Disciplines

Stefan Biffl  Alexander Schatten  Edgar Weippl
Christian Doppler Laboratory SE-Flex-AS
Institute of Software Technology and Interactive Systems (ISIS)
Vienna University of Technology
http://cdl.if.tuwien.ac.at
Agenda

- Motivation
- Integration Challenges
- State of the Art
- Use Cases
- Research Approach (Open EngSB)
- Paradigms
- Technical Integration and Security
- Planned Results
Motivation and Overview

1. Observed trends in software-intensive systems development

2. Surprisingly little work on the flexible, efficient, and robust integration of engineering tools across engineering disciplines.
Goal

- Concept & tools for agile (software+) engineering environments.
- Introduction of “Engineering Service Bus” (EngSB)
- Concept evaluation based on real-world use cases and prototypes
- Focus on Technical Integration of Tools
Integration Challenges and Requirements
Requirements for the Integration Solution

- Vendor-neutral: why?, Open Source
- platform-neutral;
- “Science Friendly”

- tailororable
- incremental introduction process.
State of the Art

- **Vendor-specific integration** in automation engineering (e.g., Comos PT)
- IDE-based point-to-point integration concepts (e.g. Eclipse)

- **Enterprise Service Bus (ESB)**: generic model, needs adaptation to automation software engineering environments.
- Service Oriented Architecture
Our Previous Work in this Context

- Tool integration studies in **Air Traffic Management** domain (Frequentis).
- **Event-Driven Architecture** (Middleware)
- **Technical integration** concepts for automation/software engineering environments, derivation of ESB configuration.
Defect detection in the Engineering Process
State of the Art

- Methods for defect detection in general software engineering:
  - Artifacts inspection
  - Model checking
  - Testing
  - Test-first development
- UML model versioning and conflict detection

→ Adaption for Automation Engineering necessary
Defect detection in the Engineering Process: Our Previous Work

- **Software defect detection and prediction methods and models**
- Value- and risk-based **software test planning**
- **Test-first software** development for automation systems
- Test management & simulation for **production automation system**
Research Approach – Automation Service Bus

1. **Design Time**
   - **Project Manager**
   - **Process Engineer**
   - **Elec. Engineer**
   - **Automation Service Bus Support**
     - **Requirements Management**
       - Tool Data
     - **Pipe & Instrumentation**
       - Tool Data
     - **Electrical Plan**
       - Tool Data
     - **Engineering Knowledge Base**
     - **Engineering Workflow Rules**

2. **Run Time**
   - **Automation Device**
     - Device Data
   - **Diagnosis Device**
     - Device Data
   - **Onsite Eng. Environment**
     - Tool Data
   - **Maintenance Engineer**
   - **Operator**
   - **SCADA**
     - Tool Data
   - **Data Analysis/Simulation**
     - Tool Data
   - **Software Dev. Environment**
     - Tool Data
   - **Engineering Service Bus (Offsite)**
   - **ASB (Onsite) Control Service Bus**

**Legend:**
- C: Connection
- Tools: Authoring and management of the automation service bus.
Solution Approach

Enterprise Service Bus

Core Components

Tool Domains

OpenEngSB.org Middleware

SVN Connector

Test Domain

Registry

Workflow

SCM Domain

Issue Track Domain

Message Bus

NUnit Connector

NUnit

Mantis Connector

Email Connector

Client Tool Connector
Use Case: Andritz Hydro
Use Case: SE Dependency Management in Distributed Teams

1. Test-Driven Development
2. Fast & Traceable Communication On Issues

Team Blue Develops Application A

Dependent on Team Red
Solution Approach Paradigms

- Open Standards/Protocols
  - Open Engineering Service Bus as Open Source Project

- Community Interaction
  - Engineering Communities
  - Open Source community
  - Research Communities

- Modularity

- Agility (Processes)

- Flexibility (Tools/Processes)
Technical Integration and Security

- Three levels of security integration
  1. Control: Service/User Access Control
  2. Prevention: Detailed Risk/Threat Analysis, Message-Level Security
  3. Analysis of Tool/User Interaction Patterns

- Two Usage Scenarios
  - One OpenEngSB Instance
  - Multiple OpenEngSB Instances connected via bridges (Bridge-Security)
Position of the Research Group in the International Context (Techn. Integration)

The research group is a rare case of collaboration of software engineers and industrial automation systems engineers.

Related research initiatives for cooperation and comparison of results

- **Socrates**: Service-oriented approach to integrating operational multi-vendor automation systems components.
- **Virtual Automation Network**: homogeneous communication infrastructure for multi-site production facilities.
- **Eclipse Jazz/ALF**: Application Life cycle Management Frameworks.
Summary Technical Integration

- Tool integration across engineering domains

- Foundation to improve
  - Engineering processes
  - Process analysis and improvement
  - Defect detection

- Kernel and concepts: open source/protocol
Dr. Alexander Schatten

- **Univ.Ass. Institut Softwaretechnik & interaktive Systeme**
- **http://www.schatten.info**
- **Forschungs- und Anwendungsgebiete**
  - Software Engineering
  - Open Source Technologien (Java)
  - E-Commerce & Enterprise Architectures
  - Event-Driven Systems
- **Kooperationen:**
  - Frequentis (Semantische Datenintegration & Optimierung)
  - Österr. Computer Gesellschaft, Leitung des Open Source Arbeitskreises, Mitorganisator Informatik
  - Consulting, Training, Vorträge (Kunden u.a. IBM, GFT, VOEST Linz, CSC, IQSoft, Softwarepark Hagenberg, Landesregierung Salzburg, Joyn IT, Steiner-HiTech, Erste Bank Leasing, Indoqa, LOVO, ...)
  - MTEL Bulgarien (Mobilkom Tochter)
  - TU Prag, Department of Cybernetics; Prof. Marik