SIM-TBASSCO: Semantic Interoperability Measures: Template-based Assurance of Semantic Interoperability in Software Composition

Project Goals

A goal of the SIM-TBASSCO project at USC ISI is to develop a metadata framework for describing software components that supports the dynamic assembly of software systems and helps users overcome software failures/bottlenecks. The framework provides multi-level architecture views that enable dynamic, rapid response to faults by providing visibility into the systems and by identifying control points for adjusting their behavior. The use of multi-level views widens the range of adjustments available to users.

As a testbed application for this work, we have adopted GeoWorlds, a component-based system for managing Web and geographic information. GeoWorlds is in operation at US Pacific Command (USPACOM), where it is used by analysts at the Virtual Information Center, as well as by USPACOM’s Crisis Operations Planning Team. Using our semantically-based scripting tool, application developers create scripts of special-purpose information analysis applications by “wiring together” components from the GeoWorlds component library. Later adjustments to the applications can be applied both at the dataflow architecture level and at the system architecture level. At the dataflow level, semantic gauges identify alternative compatible software components as candidates for substitution in application scripts. Dataflow level substitutions are dynamically applied at the system level during runtime. The dataflow style avoids the need to rerun entire scripts after modification by allowing partial recomputation based on dependency analysis. At the system level we demonstrate adjustments that apply semantically invariant transformations (such as connector transformations and component redeployment transformations). These transformations modify the system level architecture to improve performance, while maintaining the semantic constraints imposed by the scripts.

Faults Overcome via Semantic Adaptation and Semantically-Invariant Transformation

Dataflow Architecture View

- Load dataflow architecture; extend it at run-time
- Update dataflow architecture to replace malfunctioning service at run-time

Model Adaptation: Run-time Service Failure Due to Host Crash

Semantic Adaptations

- Acme dataflow architecture view depicts connections and dependencies among component services for the application
- Multiple architectural views of application to enable monitoring and analysis of distributed systems
- Dynamic view of execution progress: overall application and individual components
- Dynamic view of distributed system environment: hosts, servers, and jobs
- Semantic adaptations adjust the application
- Semantically-invariant transformations adjust the system environment
- Find alternative service types
- Modify dataflow
- Automatically detect overloaded server
- Automatically update system architecture to reflect re-hosted service
- Migrate servers to new host

System Architectural View

- Acme system architecture view depicts connection hierarchy among GeoWorlds component servers
- Detect overloaded server
- Automatically update system architecture to reflect re-hosted service
- Migrate servers from the Overloaded Host
- Model Transformation: Migrate Servers from the Overloaded Host (99% speedup of architectural revision; hours to seconds)

http://www.isi.edu/tbassco

Distributed Scalable Systems Division, USC Information Sciences Institute
**Testbed Application: GeoWorlds**

Geo-spatial Information Management System

- Large component-based system, in experimental use at PACOM & JFCOM
  - Supports crisis ops planning and execution
  - PACOM and JFCOM Joint Futures Laboratory serve as outside evaluators

GeoWorlds is a, “significant capability... that helps us to rapidly find, filter and organize information specific to locales and topics of concern to us.”

-- Jens Jensen, Chief, Operation Planning Team, USPACOM

**IntelliGauge TIE: DASADA Applied to Internet Information Systems**

Goal: Improve Internet Information Systems Throughout the Software Lifecycle

- Adapt System to Overcome Failures and Bottlenecks
  - ISI: Dynamically Modify Scripts for Service Substitutions
  - ISI: Transform Architecture while Preserving Script Semantics (Redeployment, Alternate Connectors, etc.)
  - CMU: Reconfigure Architecture (Tailor)
- Automate Checking of Component Interoperability to Reduce Time to Build Information Management Applications
  - CMU: Model Component Behavior and Data Using Acme Language
  - ISI: Generate Semantically-based Application Scripts
- Monitor & Detect Architectural Misbehaviors and Failures
  - ISI: Monitor Application’s Current Behavior
  - OBS: Determine Application’s Normative Behavior
  - CMU: Verify Structural Constraints (Armani)
  - Columbia: Verify Dynamic Behavior (XUES)
- Automate System Adjustments Based on Current Conditions
  - BBN: Adapt Workflows for Service Deployments and Migration

**Build and Evaluate High-Speed Maintenance and Extension Tool Suite**

- Combines Geographic Information Systems and Web processing services
- Architected from the ground up as a component-based framework
- Presents key common software challenges
  - Assembling custom analyses from components
  - Restructuring new version releases
  - Integrating new functionality
  - Adapting for local environments
- Detecting and resolving service failures, bottlenecks, and network congestions

For More Information:

Robert Neches, Ke-Thia Yao, In-Young Ko, Robert MacGregor

Distributed Scalable Systems Division
USC Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292

tbassco-local@isi.edu
310/822-1511

Distributed Scalable Systems Division, USC Information Sciences Institute