Context-Driven Adaptive Monitoring for Supporting SOA Governance

Norha M. Villegas
Hausi A. Müller
Rigi Group - Computer Science Department
University of Victoria, Canada

nvillega@cs.uvic.ca
http://webhome.csc.uvic.ca/~nvillega/

hausi@cs.uvic.ca
http://webhome.cs.uvic.ca/~hausi/
Service-oriented applications are highly dependent on environmental information

How can we represent the context that can affect the behavior of a service-oriented system?

How can we change this representation at run-time as required by changes in business objectives and the environment?

How can we manage the relevant context under changing requirements?
Outline

1. The research problem and the promise of service-oriented computing
2. Challenges in dynamic monitoring for SOA governance
3. Our contribution
4. Application
5. Our Approach
6. Ongoing research and summary
The Research Problem

Dynamic Monitoring of Service-Oriented Systems
Service-Oriented Computing: its visionary promise

- Creating dynamic and agile business processes
- Spanning organizations and business platforms

By assembling components into loosely coupled network of services

Complex Dynamics in Service Composition

- Service composition must understand and respect or is affected by:
  - Policies
  - Performance levels
  - Security requirements
  - Service level agreements
  - User’s matters of concern

Dynamic service composition is affected by environmental information that characterize the situation of users, services, computational platforms and businesses.

Context Monitoring is Key!
Some MESOA Research Challenges

- Design for context-awareness
- Automated governance
- Run-time monitoring
- Dynamic service discovering
- Dynamic service composition
- Adaptive maintenance and evolution practices
- Mapping between business process and services
- Run-time V&V (compliance with design and run-time constraints)

Dynamic Context Monitoring for SOA Governance: Research Challenges

SOA governance needs to manage context information to ensure the execution and evolution of service-oriented systems.

Relevant context must be specified at design-time but managed at run-time.

Monitoring requirements are dynamic, the relevant context is not the same over time.

- Business objectives
- User’s concerns
- New service interactions
- Different SLAs
- Distributive nature of services
Dynamic Monitoring Instrumentation

Service-oriented monitoring infrastructures are required to adapt themselves to address dynamic monitoring objectives according to changes in business-level objectives and processes, user’s concerns and the environment.
Optimizing SOA Governance

SOA governance requires effective mechanisms to control the accomplishment of governance objectives under changing environmental conditions.
Our Contribution: Dynamic Monitoring for SOA Governance

- A feature-based model to represent relevant context and monitoring requirements
- An adaptive control-based reference architecture to implement dynamic context monitoring infrastructures
- Feedback loops are first level components
- Monitoring objectives keep relevance with respect to governance objectives
- New monitoring strategies defined at run-time
Application

Through Run-Time Re-Negotiation of SLAs
Run-Time Re-Negotiation of SLAs

SLA: Minimum transaction rate
SLO: throughput

Minimum transaction rate

10 trans/sec → summer
5 trans/sec → the remainder of the year

SCA Artifacts Legend

Promoted references
Reference
Component
Property
Composite

HotelNearby Facilities

Shopping Facilities Broker A

SpecialOffers StoreA

Tourism Facilities Broker A

SpecialOffers StoreB

Shopping Facilities Broker B

SpecialOffers StoreC

SpecialOffers StoreD
Run-Time Re-Negotiation of SLAs

The dynamic re-negotiation of SLAs should be performed according to current situations:

Changes in the hotel occupancy rate
Run-Time Re-Negotiation of SLAs
A Control-Based Reference Architecture for Implementing Dynamic Context Monitoring

Governance Feedback Loops
Feature-Based Context-Driven Context Management
Context-Aware Governance Feedback Loops

Feature-Based Context Meta-Model

1. Our context taxonomy

- Individual
  - Natural
  - Human
  - Artificial
- Location
  - Groups of Entities
- Time
  - Definite
  - Indefinite
- Activity
  - Social
- Relational
  - Functional
  - Compositional
- Context Classification

Minimum transaction rate:
- 10 trans/sec → summer
- 5 trans/sec → the remainder of the year

Feature-Based Context Meta-Model

2. Features to guide context modeling and management

Control-Based Service Component Reference Architecture
Control-Based Service Component Reference Architecture
Context-Model Controller Architecture

Inference of context control objectives

Inference of monitoring strategies

Definition of monitoring plans

Monitoring Plan
Context Monitor Architecture

Monitoring Plan  ➔ Deployment of the strategy ➔ Gathering ➔ Handling ➔ Correlation of context facts ➔ Symptoms
Context Sensing Architecture

Composite: Sensing Infrastructure

Context sensing request → Context gathering → Sensor discovery → Pre-processing → Context observation
Dynamic Monitoring Strategies

![Diagram showing the flow of data and interactions between different components such as Governance Infrastructure, SLO Monitor, Transactions Handler, DateTime Handler, Transactions Acquisitor, DateTime Acquisitor, Occupancy Handler, and Sensing Infrastructure. The diagram illustrates the process of monitoring transactions, handling dates and times, and integrating occupancy data.](image-url)
Case Application: Concrete architecture for the monitoring infrastructure required by the case application
Related Work

• Current SOA monitoring approaches are not context aware (e.g., IBM-WSLA)

• There is not many evidence of monitoring approaches able to change dynamic monitoring strategies at run-time (e.g., approaches surveyed by Bianco et al., Villegas and Müller, and the Petals European Project)

• Aspect-oriented approaches provide partial dynamicity (e.g., Dynamo of Baresi and Guinea’s)
  ▫ Monitoring assertions are generated at compilation-time or load-time
  ▫ Once at run-time, logic monitoring is statically integrated into business logic

• Current approaches focuses on requirements for the adaptation of the managed system rather than on the adaptation of the monitoring strategy itself
Ongoing Research

- Definition of case studies from industry to validate the approach (e.g., IBM-CAS)
- Evaluation of the feature-based model approach to represent context and monitoring requirements
- Evaluation of available technology to support the modification of monitoring plans at run-time
- Evaluation of existing context management frameworks

- Design-time representation
- Run-time modification
- Support for user modification
- Apache Tuscany
- IBM WebSphere application server
- FraSCAti
- COSMOS
Summary

- We proposed a feature-based reference model to guide the representation of context information and monitoring requirements for SOA governance
  - Represented at design-time
  - Managed at run-time

- We proposed a control-based reference architecture to guide the implementation self-adaptive monitoring infrastructures
  - Based on feedback loops to control the adaptation process
  - Able to implement dynamic monitoring strategies deployable at run-time according to SOA governance objectives

- The next steps focuses on
  - The validation of models to represent context at design-time and then manage it at run-time
  - The validation of available technologies for implementing dynamic monitoring strategies
  - The implementation and validation of the monitoring infrastructure to be applicable in industry
Thank you!

Questions and discussion

University of Victoria, aerial view