Service Orchestration: It takes more than XML-based standards

Mike Smit
March 4, 2008
The Orchestration Problem

- Even “correct” service compositions “fail” sometimes
  1. data management
  2. control management
  3. performance

- What is the role of reflective orchestrating middleware and runtime monitoring?
Service Composition 101

- **Service**: a network accessible software component, aligned with a business function
- **Composition**: a hierarchical organization of services, accomplishing a higher-order business process and function
- **Intra- or inter- enterprise**
  - More interesting when linked services are developed independently
Orchestration Concerns

1. **Data management:**
   - if Mike has given his credit card number to WestJet, is it OK for Delta to also have it?

2. **Business Protocol Concerns:**
   - What if Visa requires a “signature” from Mike?

3. **Performance:**
   - How does Visa ensure that it meets its SLAs without over-reserving resources?
1) Data Management: Privacy & Security

- **The context**
  - Orchestrations are subject to laws, standards, internal policies, published policies
  - These regulations vary by jurisdiction and domain

- **Case in point**
  - TJX, ChoicePoint, Hershey’s
Hershey’s Website – After COPPA settlement
Run-time Monitoring for Data Management: Privacy & Security

- The research problems
  - How do you represent regulations?
  - How do you recognize the regulations relevant to an orchestration, especially under the assumption that orchestrations can change dynamically?

- The approach
  - Think of it as distributed database transaction?
    - Some communication is peer-to-peer
  - Formal Methods?
    - Orchestrations are dynamic
  - Data Flow Analysis
2) Business Protocol Concerns

- **The context**
  - Each organization, contributing services to a composition, is subject to its own internal rules and protocols
    - Which may evolve independently

- **Case in point**
  - After FOIPP, instructors cannot request student transcripts; students have to request student transcripts to be sent to instructors
Run-time Monitoring for Business-Protocol Negotiation

- **The research problems**
  - How do you represent independent protocols?
  - How do you recognize conflicts?
  - Which conflicts can be resolved automatically?

- **The approach**
  - The unit of service modeling should be “service provision requires” protocols (abstract BPEL)
  - Organization-specific agents monitor orchestrations, based on these normative specifications
  - Divergences are recognized, negotiated and resolved among agents, based on shared policies, based on
    - Org chart, contract
    - Cost efficiency
    - Randomly

http://www.cs.ualberta.ca/~stroulia/wrabbit
3) Performance

- **The context**
  - Interaction between services can do unexpected things to performance
  - Bottlenecks, failure & recovery, latency…
  - Configure to meet SLAs… not a static environment

- **Case in point**
  - ???
Deployment configuration for Performance

- The research problem
  - How to represent SLAs?
  - What does it mean to “fail”?
  - What is relevant to model?

- The approach
  - Model
    - Network topology, capacity
    - End-points capacity
    - Traffic
    - Recursively simulate
  - Optimize
    - Minimize capacity
    - Avoid violations
Run-time Monitoring for Performance

- **The research problems**
  - Autonomic deployment-configuration adaptation

- **The approach**
  - Recognize actual behavior as similar to “failed simulation”
  - Modify provisioning as per simulation plan
Run-time Monitoring

- Broadly, a reflective middleware solution
  - Different components address the various problems
  - Comparing a set of desired attributes (privacy policies, data flow, control flow, performance) to actual behavior
    - … and ideally, correcting actual to match desired
Questions / Answers