Service Composition

Panel discussion

Dr. Schahram Dustdar
Associate Professor

Distributed Systems Group
Vienna University of Technology
dustdar@infosys.tuwien.ac.at
http://www.infosys.tuwien.ac.at/Staff/sd

ICSoC 2003

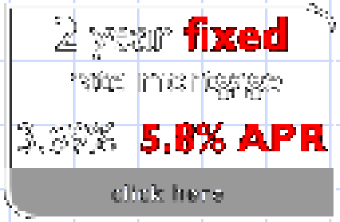
Position

- SOA = Revolution but
  Web services = Evolution
- Vendors will add extensions to the "application server"=bus=OS (sooner/later)
- Service Architect (User vs. Provider)
- SOAPY and Slippery Services
  Composition vs. Specification
- Open Source Services (Portals to Freedom?)

Forget the travel scenario

Lowest

- How to choose the mortgage that is right for **me**?
- The mortgage is just a component of a larger “**system**”: what about the insurance policies that protect the lender? what about complementing repayment with a savings scheme? what about…
- This system is **dynamic**: interest rates will change; my status will change; will I have to go all over the search process again? How many times?


2 year **fixed**
rate mortgage
3.69% **5.8% APR**
click here

Service Infrastructure

- Web browser is part of OS?!
- "Application server" is part of OS?!

- Who provides service infrastructures?
- What happened to the P2P architectural style of Web services?

Sounds familiar ?

Service Architect

- Different user/provider views
  (interface, (business-) protocols, Transport, Implementation)
- Service Evolution (new, disappear, morph; e.g. bidding for services)
- Metrics and Criteria for monitoring and managing services
- Planned vs. Unplanned change (design-time vs. Run-time?)
- Service (Process) reuse
- Service simulation and service testing

SOAPY Services

- Need to be monitored and maintained (notifications to who) in run-time to reflect **changes** on
  - the way the service is being delivered (e.g. due to increased mobility -> connectivity modes)
  - the business environment (e.g. new legislation)
  - the requirements (e.g. client status changes)
  - the technological opportunities (e.g. more robust components are made available)
- Missing:
  - support for Evolution
  - support for Configuration
  - support for Life-cycle and versions
  - …

Business Models

- Software companies jointly provide new applications (compositions?) consisting of their services

- Open Source Services as Portals to Freedom?

- (If) services are free; What about the process composition (templates)? Who adds value to services?

Thank you for your attention!

Compostion Modeling aspects

- Composition (BPEL, WS-CAF ...)
- Orchestration (UML, *.calculus, Petri nets, ...)
- Data (who does data transformation, how)
- Selection ((de)central/federated registries)
- Transactions (defining the scopes)
- Verification, Correctness etc.

Plenty of opportunities to add proprietary extensions to the application server = middleware = OS...

Service oriented Economy...

- Computing is becoming a utility and software a service. This will profoundly change the economics of the IT industry.
- Applications will no longer be a big chunk of software that runs on a computer but a combination of web services; and the platform for which developers write their programs will no longer be the operating system, but application servers.

IT industry survey "The Economist" (10 May 2003)

Service vs. Components

- More than a terminological issue…
- Shift from server-to-server, static, linear interaction based on identities to dynamic, mobile and unpredictable interactions based on properties (**who** vs **what**);
- “late” or “just-in-time” integration (as opposed to compile or design time integration): interconnections need to be established and revised dynamically, in run-time, without interruption of “service”.
- Design elements to which attributes like **quality** and **trust** are most frequently assigned.

Web Service Composition

- What is Composition really?
  1. Composition in the “part-of” sense (granularity),

     i.e. larger part encapsulates web-services (composite) and exposes itself as a web-service

     Analogy: method invocations as part of method definition
  2. Composition in the “sequencing” sense,

     i.e. definition of the invocation order of web-services (often called orchestration or coordination)

- **Working Definition:** Composition consists of those activities required to combine and link existing Web Services (atomic and composite services) and other components to create new processes.

Web Service Composition

- Types of Composition
  - Static Composition - services to be composed are decided at design time
  - Dynamic Composition - services to be composed are decided at run-time

- Some languages for web service composition
  - BPEL4WS (Business Process Execution Language for Web Services)
  - WSFL (Web Services Flow Language)
  - XLANG (BizTalk)
  - BPML (Business Process Modeling Language)
  - ebXMl BPSS (Business Process Specification Schema)

Composition Issues

- Representation of an abstract Web Process
  - Representing/specifying the abstract process in a proper form
- Discovery and Interoperability of Services
  - Need to manually or automatically search for appropriate services
  - The discovered services should interoperate
- Efficiency of a Composed Web Process
  - Need to compose processes which are efficient in terms of performance
- Process Execution
  - Adopting a suitable technique for executing the composed concrete process
- Process Monitoring
  - Using a monitoring technique for run time analysis of the Web process execution

Questions

- Service selection – How and Who?
  - Static binding
  - Dynamic binding by reference
  - Dynamic binding by lookup
  - Dynamic operations selection
- Dealing with service specific states: data flows during WS conversations?
- Transformation of data between services (WS-Mediation)?

Questions

- Services are aggregated to master complexity; for whom?
- At least 2 views are needed
  - User View (Registries, Granularity, Policies)
  - Provider View (Policies, Billing, Controlling)
- Composition vs. Specification
- Reuse of process compositions