

MWS 2007 Discussion Session

“Impact of Various Execution Environments on Middleware for Web Services”

Moderator: Vladimir Tasic
NICTA (Australia), UNSW (Australia), UWO (Canada)



Australian Government
Department of Communications,
Information Technology and the Arts
Australian Research Council

NICTA Members

Department of State and
Regional Development

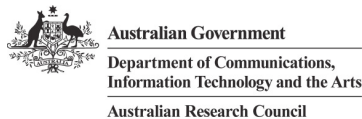
**NICTA Partners**

About NICTA



Slide 2

- Australian center of excellence in use-inspired ICT (info & communications technology) research
- Established 2002, now 5 labs in 4 cities
- NICTA members (governments and universities):



Department of State and Regional Development



- NICTA partners:



The University of Sydney



Queensland Government



Queensland University of Technology



THE UNIVERSITY OF QUEENSLAND AUSTRALIA



- 422 research staff, 296 postgraduate students
- 4 research themes (e.g., managing complexity) & 6 business areas (e.g., software infrastructure)

Various Execution Environments

- Business server-to-server (traditional)
 - Mobile (mobile requesters and/or providers)
 - Embedded and/or ubiquitous
 - Grid
 - Data centers?
 - Desktop servers?
 - ... (participants add)
-
- Usage of WSeS in various environments increases

Some Environment Specifics

- Mobility
 - Location changes
 - Context (location, relative distance, ...) awareness
- Embedded systems
 - Limited processing power
 - Limited memory
 - Limited communication bandwidth
 - Limited electrical energy
- Grid
 - Virtualization
 - Instance life cycle management

Motivation for the Discussion

- Due to environment specifics, middleware for WSeS executing in one environment might need significant modifications for other environments
 - Example: Location changes of mobile requesters (providers) can be handled with proxies (gateways)
- Many MWS 2007 participants have significant experience about these issues and their impact
- However, there are still many unknowns
- Also, these issues are often not formally identified and discussed
 - This leads to some repetition of past work

Goals of the Discussion

- Identification of issues
- Assessment of their impact
- Exchange of information about possible solutions
- Exchange of information about non-solutions
- ... (participants add)
- Other exchange of knowledge, experiences, ideas, and opinions

Requirements for Mix-and-Match

- There should be high uniformity of how WSEs are handled (e.g., accessed, managed) irrespective of their execution environment
 - Necessary for porting existing solutions to new environments (ideally, in an automatic way)
 - The uniformity need not be complete, due to the environment specifics
- It should be possible to compose WSEs executing in various environments (e.g., mobile and static)
 - Necessary for maximizing benefits and overcoming limitations (e.g., in resources)

Requirements for Adaptability

- Middleware should automatically adapt (or at least can be adapted by users) when WSEs change the type of execution environment
 - Necessary for dynamic evolution of WS systems
- Parts of the middleware could also run in various environments (e.g., mobile vs. static) and the middleware should be able to reconfigure itself
 - Necessary for minimizing overhead, overcoming limitations (e.g., in resources), maximizing benefits
 - Overhead of any middleware should be minimal

Impact of Mobile Environments

- Additional middleware communications over wireless links should be minimized
 - Wireless is slow, unreliable, and eats batteries
 - Processing messages in the stationary network
 - Piggybacking is better than separate messages
- Disturbances (relatively frequent) have to be handled quickly and with limited resources
 - Wireless is more error-prone, coverage is not equal, there are sleeping phases
 - Must be handled at several levels: communication, application (e.g., contract adaptation)

- Middleware could provide store-and-forward of messages to temporarily unavailable WSEs
 - Configurable: timeout, retry interval, maximal number of retries, availability notification flow, ...
- (Stationary) Caching, pre-fetching, cache management, and lookup could be useful
 - Also saves resources, but not always applicable
 - Required information: cache expiration or duration, cache invalidation conditions, ...
- Middleware could enable opaque exchange between replicas (special case of re-composition)
 - Also useful for load balancing

- Middleware should enable publishing, monitoring, storing, processing, analyzing, communicating, and updating context information
 - Requires specification of context and its impact
- Not only WSEs, but also middleware operations (e.g., WS management), should be context aware
 - E.g., monitoring, accounting, billing, analysis, planning, and control activities
- WSEs, WS compositions, and middleware should automatically adapt to changes in context