Making Web Services Dependable

L.E. Moser, P.M. Melliar-Smith and Wenbing Zhao
International Conference on Availability, Reliability, and Security (ARES’06).

Present By
Gaspar Modelo Howard
Ratsameetip Wita

DCSL Reading Group 12/04/2007

Outline

• Introduction
• WS Dependability Specification
• Dependability in WS Architecture using fault tolerance technique
• Conclusion
Introduction

- Basic Web Service Standard
  - eXtensible Markup Language (XML)
  - Simple Object Access Protocol (SOAP)
  - Web Service Description Language (WSDL)
  - Universal Description Discovery and Integration (UDDI)

Potential widespread usage of WS

Figure 1. Use of Web Services in business-to-business activities that span multiple enterprises.
New problems rising

- One transaction involves multiple WS.
  - Multi-enterprise business activity.
    - One participant can effect another enterprise.
- One WS involves multiple components.
  - All must be dependable.

![Diagram of web services architecture]

High availability issue

- Business activity is comprising of multiple web services.
- If one of involving web services unavailable, all of the others are affected.
- Availability of business process << availability of any component.
- Let $n =$ # of tiers in WS architecture within enterprise
  $m =$ # of WS of different enterprise involved in business activity
  $p =$ probability of any tiers in enterprise fail
  $(1-p) =$ probability that they do not fail
  Probability of the whole business activity that not fail
  $q = (1-p)^{mn}$
High availability issue

- For $l$ independent business activity, probability of all of them are not fail is
  \[ r = q^l = (1-p)^{mn} \]

Data consistency issue

- Detecting and correcting inconsistency are difficult, time consuming and expensive due to spanning to multiple business activity.
  - Three-phase commit protocol/ replicated coordinator
- Multiple local transaction with compensating transaction is currently used in business activity.
  - High error rate and high risk of data inconsistency
- Prevent both locking of data from failed transaction and potential inconsistency from incorrect compensation are essential for business activity
WS Dependability Specifications

- Reliable Messaging
  - WS-Reliable Messaging and WS-Reliability specification.
    - Application-level reliable messaging protocol over SOAP.
    - Acknowledgement and retransmission with different QoS, e.g. at least once, at most once, exactly once, source order delivery.
    - Lack of topics of message persistence and fault recovery.

- Transaction and Business activity
  - WS-Transaction
    - Protocol for atomic distributed transaction commitment
    - Based on two-phase commit protocol (2PC)
    - Provide data consistency
  - WS-Business Agreement
    - Support long-running transaction
    - Determine business activity should roll forward or roll back
  - WS-Coordination
    - Describes an extensible framework for providing protocols that coordinate the actions of distributed applications.
    - Strict consistency/ proper subset of participant

Figure 4. Reliable messaging protocol stack.
Multi-component Web service

![Diagram of Web Services architecture](image)

**Figure 5. Three-tier Web Services architecture.**

Dependability in WS Architecture

- Can be achieved using fault tolerance technique.
  - Application logic
    - Replication on Web sever, Servlet, J2EE Application server
  - Transaction Coordinator
    - Non-blocking 2PC protocol
    - Exactly-once semantic for clients’ invocations.
  - Database Server
  - Web Service Registry
  - Business Activities
    - Extension of WS-Coordination and WS-Business activity specification
    - Reservation-based coordination protocol (by same author)
Fault tolerant Web server

- Building on top of WS-RMP
- Extension of RMP on recovery from fault (SOAP failover)
  - Restore WS to checkpoint state.
  - Restore TCP connection.
  - Replay logged receive message subsequently from the checkpoint.
  - Detect and suppress all delivered message during fault.

Figure 6. Fault-tolerant Web server.

Replication in Three-tier Web service Architecture

Figure 7. Fault-tolerant three-tier Web Services architecture.
Replication in Three-tier Architecture

Figure from: Moser L. et al, Unification of Transactions and Replication in Three-Tier Architectures Based on CORBA, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 2, NO. 1, JANUARY-MARCH 2005
Reservation-based coordination protocol

- Avoiding compensation transaction.
- Extension of WS-Transaction specification.
- Two-phase protocol
  - Reservation phase
    - Set involved resources as reserved
    - Coordinator determine reservation confirm/cancel by business rules.
  - Confirmation phase
    - Coordinator send confirmation/cancel message.

Reservation-based VS. Locking

- Application control resource reserved status.
- User-Resource Owner Agreement contract is used to determine reservation fee.
- Immediate notification when resources are reserved so application can find appropriate action.
  - For locking, application need to wait until timeout or lock is released.
Conclusion

• Describing Web service specification involving dependability.
• Possibility of using fault tolerance in web services with concerning business activity.
• Performance evaluation is needed.