Dependability in the Web Service Architecture

Ferda Tartanoglu

INRIA

ARLES Research Project

In Collaboration with :

Valérie Issarny - INRIA

Alexander Romanovsky - University of Newcastle Upon Tyne

Nicole Levy - Université De Versailles Saint-Quentin En Yvelines

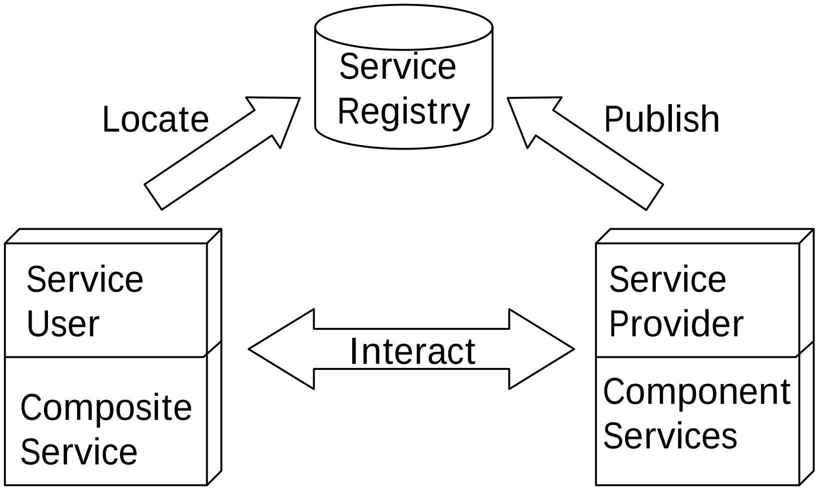
Overview

- Introduction: the Web service architecture
- Composing Web services
- Transactions for the dependable composition of Web services
- Using CA actions for the dependable composition of Web services
- Conclusions & Future Work

Introduction

- The Web service architecture targets the development of applications based on the XML standards
  - eases the construction of distributed systems by enabling the dynamic integration of applications

- Main constituents of the Web service architecture



Introduction (2)

- XML Standards
  - WSDL (Web Service Description Language, W3C)
    - a language based on XML for **describing the interfaces**
  - UDDI (Universal Description, Discovery and Integration)
    - specification of a **registry** for dynamically locating & advertising Web services
  - SOAP (Simple Object Access Protocol, W3C)
    - sets the rules of how to **encode data** in XML
    - describes **what** is in a message and **how** to process it
    - mapping to **transport protocol** (HTTP)

- Existing platforms and tools
  - .NET, J2EE, CORBA Web Services ...

Introduction (3)

- The Web Service architecture is quite recent, but
  - will play a prominent role in the development of next generation distributed systems
  - strong support from industry

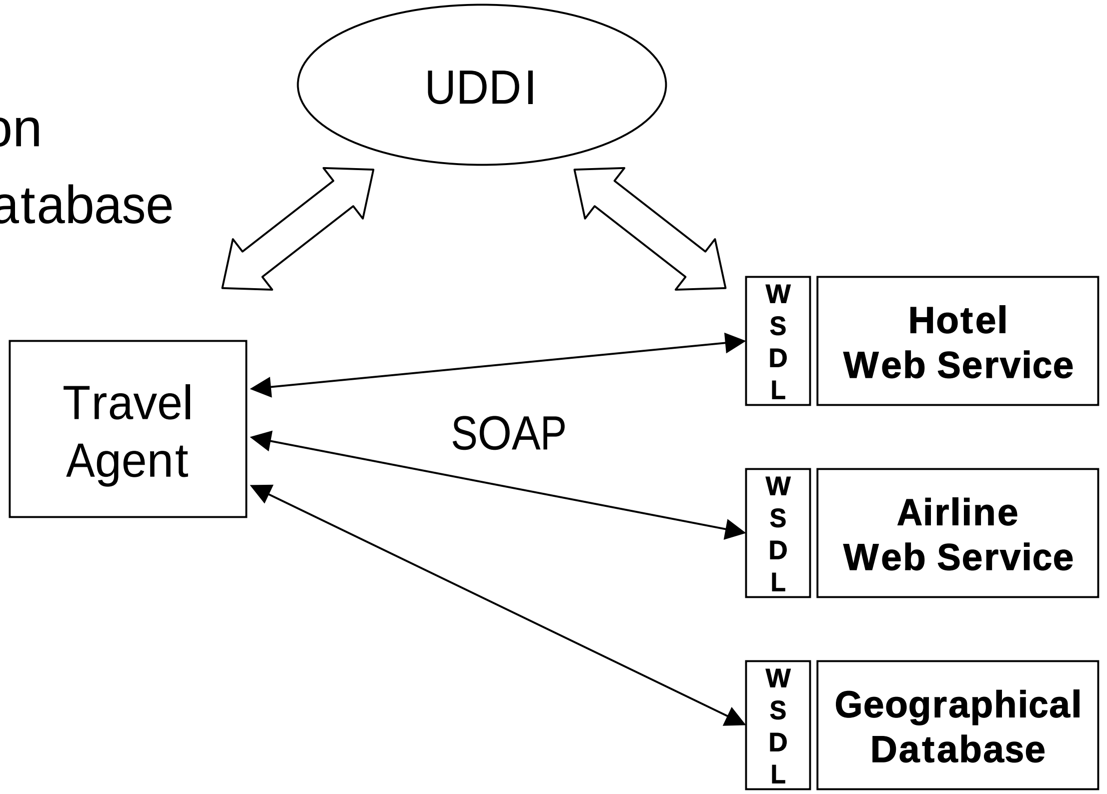
- Research challenges
  - Developing business processes with Web services
    - requires support for composing Web services in a way that guarantees dependability
    - new architectural principles needed

Composing Web Services

- Assembly of autonomous components
  - new service out of the components' primitive services given the corresponding published interfaces
- Currently,
  - interfaces are **described** in WSDL
  - and **published** through UDDI
- Supporting composition requires
  - **specification** of the composition,
    ensuring that composition guarantees consistency of
    - both the **individual services**
    - and the **overall composition**

Composing Web Services (2)

- Example : The Travel Agency
- Joint booking of accommodation and flights
  - use of existing Web services
    - hotel booking
    - flight reservation
    - geographical database



Composing Web Services (3)

- Proposals rely on a new language and supporting environment
  - WSFL (IBM)
  - XLANG (Microsoft)

✗ not yet a consensus about how the composition should be supported

- Two major trends
  - composition based on workflow management
  - using transactions to enforce dependability

Transactions for the Dependable Composition of Web Services

- Enforcing ACID properties requires introducing protocols for
  - locking resources that are accessed for the duration of the embedding transaction
  - committing transactions

✗ Such a model is not suited for making the composition of Web services transactional

Transactions for the Dependable Composition of Web Services (2)

- The management of transactions requires cooperation among the transactional support of individual Web services
  - may not be compliant with each other
  - may not be willing to do so
    - intrinsic autonomy
    - they span different administrative domains

- Locking accessed resources until the termination of the embedding transaction is not applicable to Web services
  - large number of concurrent clients that will not stand extensive delays

Enhanced Transactional Models

- The split model allows reducing the latency due to locking
  - transactions may split into a number of concurrent sub-transactions that can commit **independently**
  - it requires using compensation over committed sub transactions in the case of abortion
- Using compensation must extend to all the participating Web services
  - XLANG supports compensation operations, but
    - focus is on the behavioral spec of individual Web services
- An active area of research
  - BTP (Business Transaction Protocol, Oasis Committee)

  ✗ proposed solutions do not cope with all the specifics of Web services

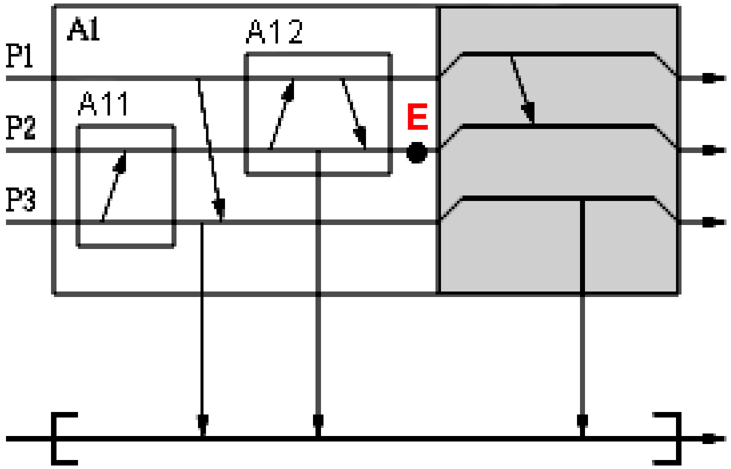
Transactions

- A major source of difficulty lies in the use of **backward error recovery** in an open system such as the Internet
  - mainly oriented towards tolerating hardware faults, **but**
  - poorly suited to the deployment of cooperation-based mechanisms over autonomous component systems
  - isolating component systems for the duration of the transaction contradicts the intrinsic autonomy of Web services
  - ✗ returning the service state back not applicable in many real-life situations

Using CA Actions for the Dependable Composition of Web Services

- Forward error recovery with Coordinated Atomic Actions (J. Xu, B. Randell, A. Romanovsky et al., 1995)
  - structuring mechanism for developing dependable concurrent systems
  - **atomic actions** : for controlling cooperative concurrency
    - coordinated error recovery using **exception handling**
  - **transactions** : coherency of shared external resources

Coordinated error recovery



Transactional object

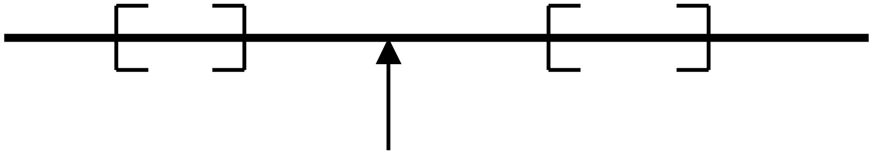
CA Actions for the Specification of Web Service Composition

- Each participant specifies the interactions with each composed Web service stating the role of the specific Web service in the composition
  - the participant specifies the actions to perform when the Web service signals an exception
    - may be either handled locally or
    - be propagated to the level of the embedding CA Action

- Each Web service is viewed as an external resource
  - ✗ unlike the base CA Action model, interactions are not enforced to be transactional

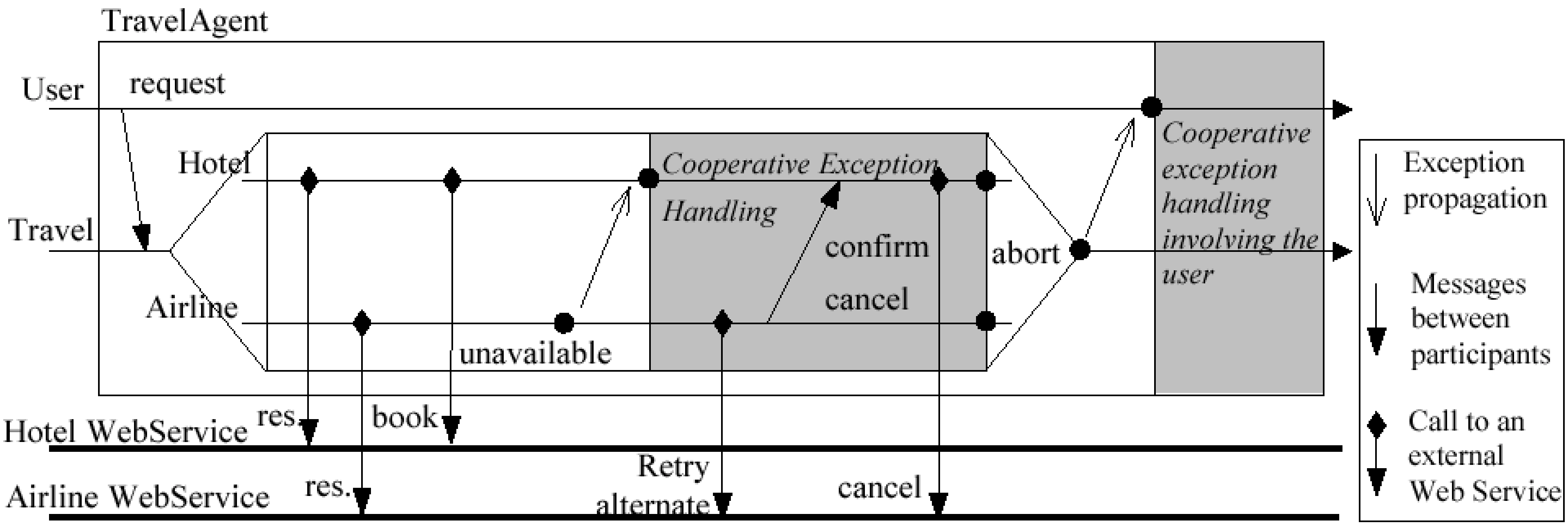
CA Actions for the Specification of Web Service Composition (2)

- The standard specification gives the expected behavior of the composed Web service
  - absence of failures
  - failures that are locally handled : no coordinated recovery
- The exceptional specification states the behavior of the composed Web service under the occurrence of failures at one or more of the participants
  - cooperative exception handling
  - the resulting forward error recovery may realize a relaxed form of atomicity



WSCA : Web Service Composition Action

- Relaxes the transactional requirements over external interactions
- Composition of WSCAs



Conclusions & Future Work

- Fault tolerance in the Web service architecture
  - use of forward error recovery
  - cooperative actions
- Dependable service composition without
  - undermining the Web service autonomy
  - increasing individual access latency
- Next step
  - formal specification of WSCAs (B formal method)
    - precisely characterize the dependable behavior
    - relaxed form of atomicity
  - architectural style