A Framework for Contract-Based Collaborative Verification and Validation of Web Services

Xiaoying Bai

Department of Computer Science and Technology
Tsinghua University
baixy@tsinghua.edu.cn

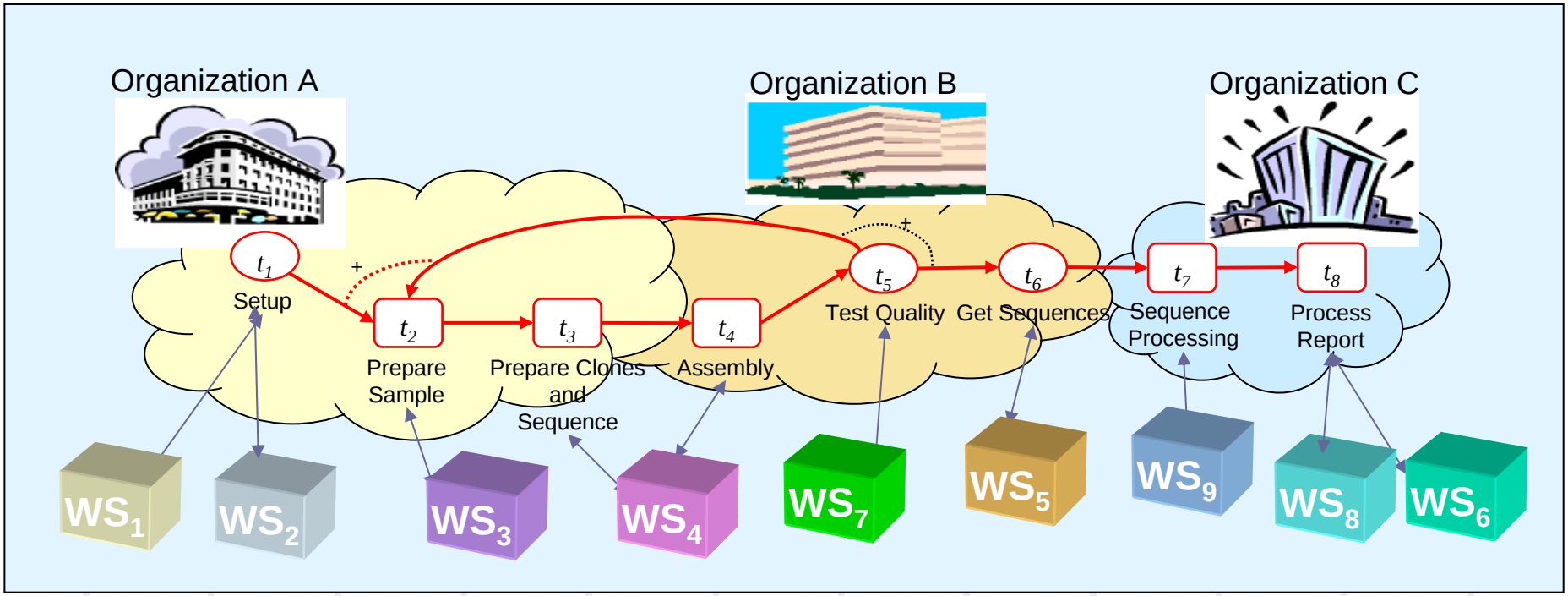
2007.7

Outline

- Research motivation and background
- The test broker architecture for Decentralized Collaborative Verification and Validation (DCV&V)
- DCV&V contracts
- Contract-Based test generation
- Conclusion

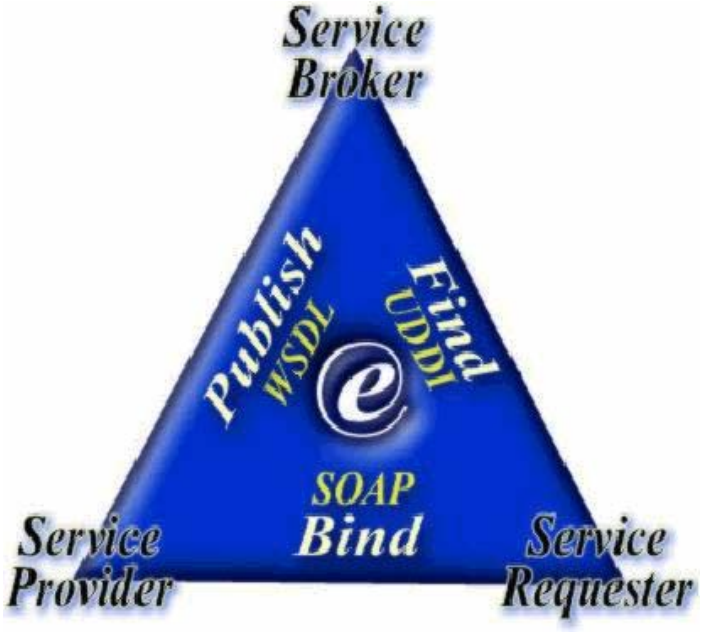
Motivation

- Service-Oriented Architecture (SOA) and its implementation Web Services (WS) are redefining the entire process of software development.
  - Software delivery, discovery, composition, collaboration



Motivation

- Standard-based dynamic collaboration is a key feature of WS systems.



???	???	Management	Quality of Service	Security
Routing, Reliability and Transaction	???			
Workflow	WSFL			
Service Discovery, Integration	UDDI			
Service Description	WSDL			
Messaging	SOAP			
Transport	HTTP, FTP, SMTP			
Internet	IPv4, IPv6			

Motivation

- Trustworthiness is a big challenge for dynamic collaboration.
  - The success of WS depends on its capability to resolve the testing issues. (Bloomberg 2002)

"*Web services are not yet widely used because of security concerns. But there's an even bigger roadblock waiting just down the road -- it's called trust. The big issue is 'Will the service work correctly every time when I need it?' As yet few are thinking about the issues of testing and certification. We suggest that testing and certification of Web services is not business as usual and that new solutions are needed to provide assurance that services can really be trusted.*"

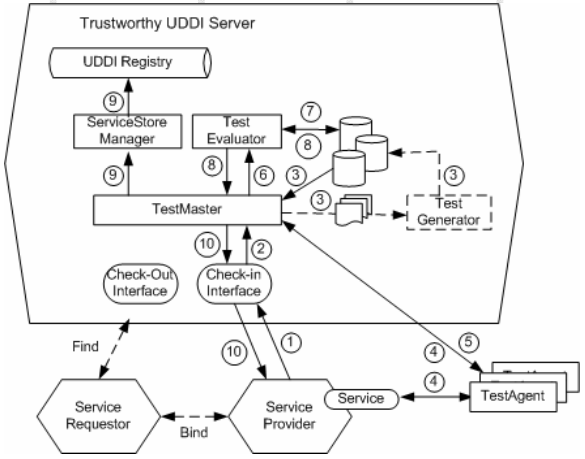
-- In CBDi Forum, 11 Jul 2002

Motivation

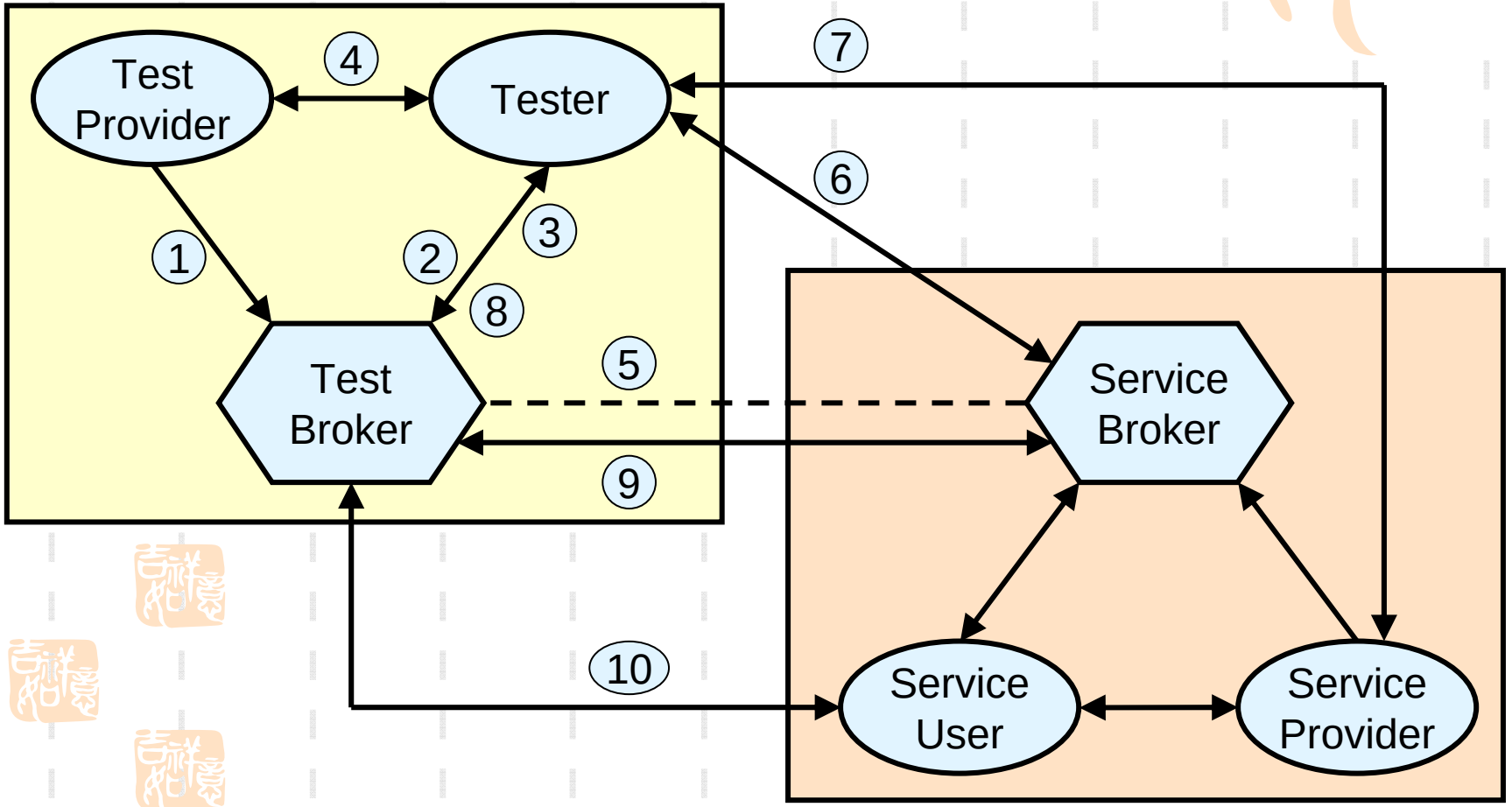
- New challenges for WS testing
  - **Assessment of a large number of services**
    - Large number of services meeting the same specification could be available over the open Internet platform
  - **Dynamic testing and automated testing**
    - The dynamically constructed application has to be tested dynamically at runtime automatically without human intervention following the specification-based approach.
  - **Cooperative testing**
    - The distributed architecture requires cooperation and collaboration among different testing activities and stakeholders including service provider, service consumer, and service brokers.

WS Collaborative Verification and Validation (CV&V)

- To enable the collaboration and cooperation of all the parties involved in SOA to perform WS testing.
- WebStrar Infrastructure for testing services and service-based applications
- The trustworthy service broker
  - Extend UDDI server by adding just-in-time WS testing, evaluation, and ranking capabilities.



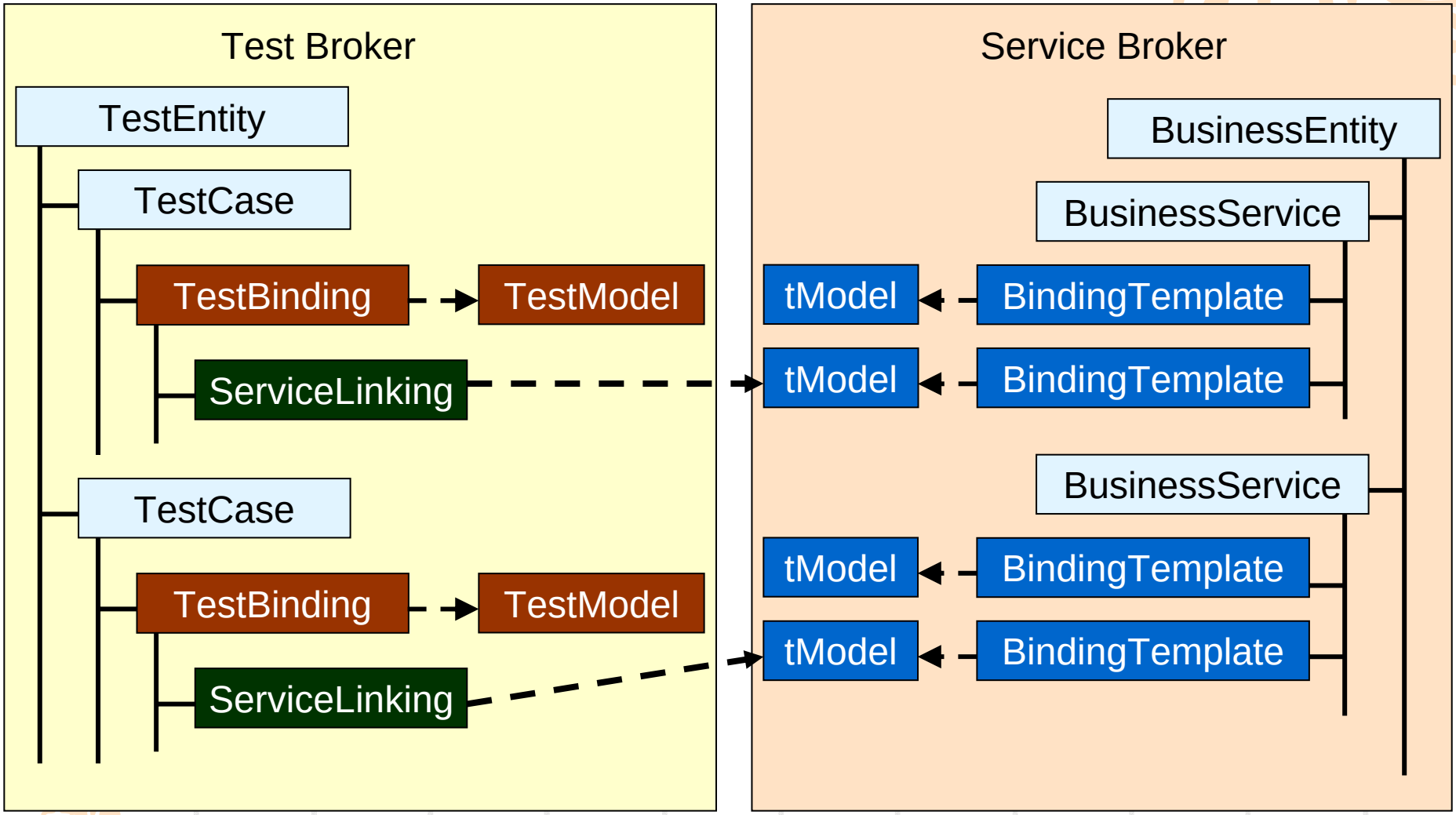
The Test Broker Architecture for CV&V



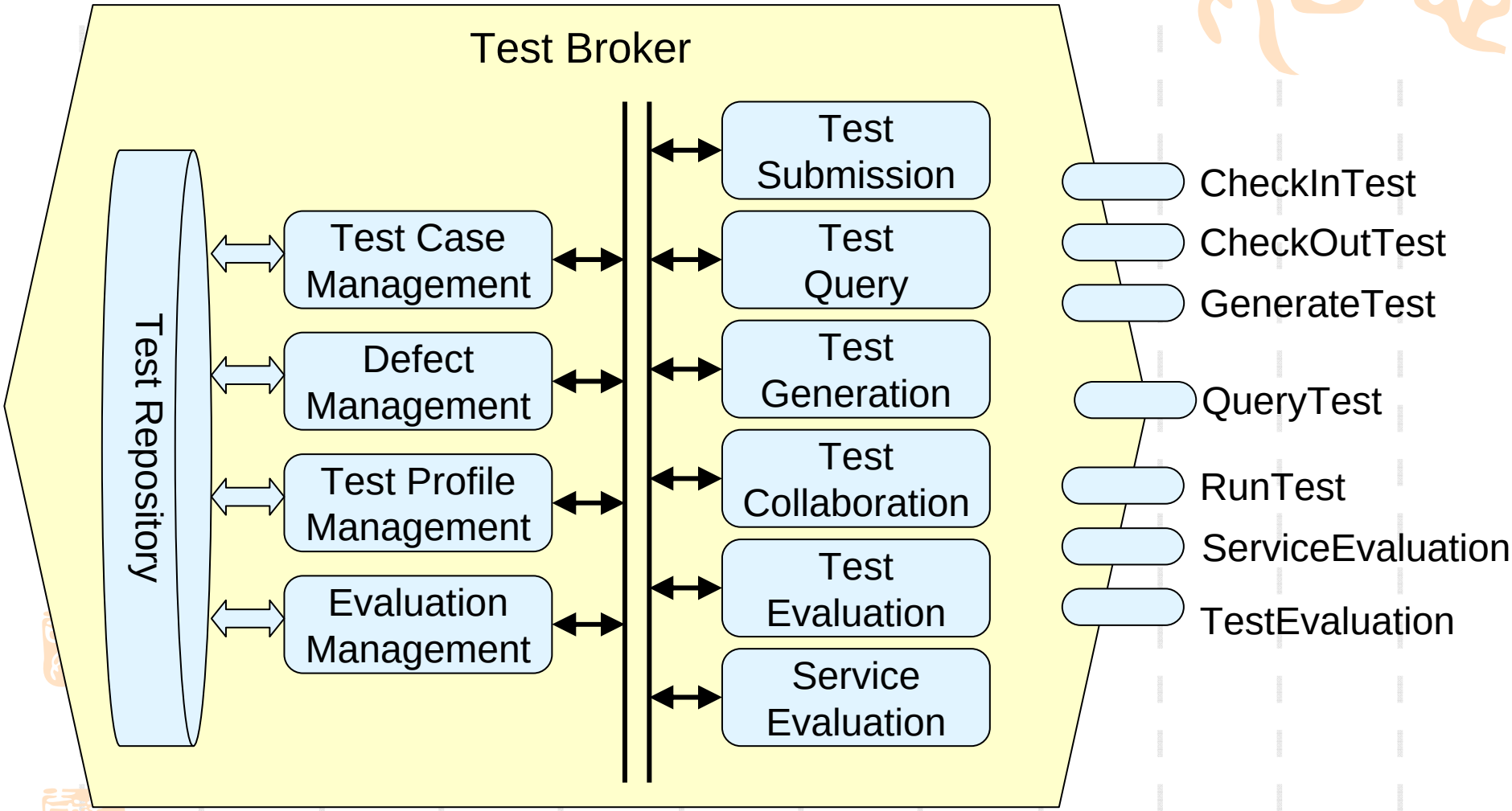
The Test Broker Architecture for CV&V

- The test provider can be
  - Service provider
  - Service requester
  - Independent tester
- The tester can be
  - Service provider
  - Service requester
  - Service broker
  - Certification organization
  - Independent tester

The Test Broker Data Structure



The Test Broker Services



Decentralized Test Brokers

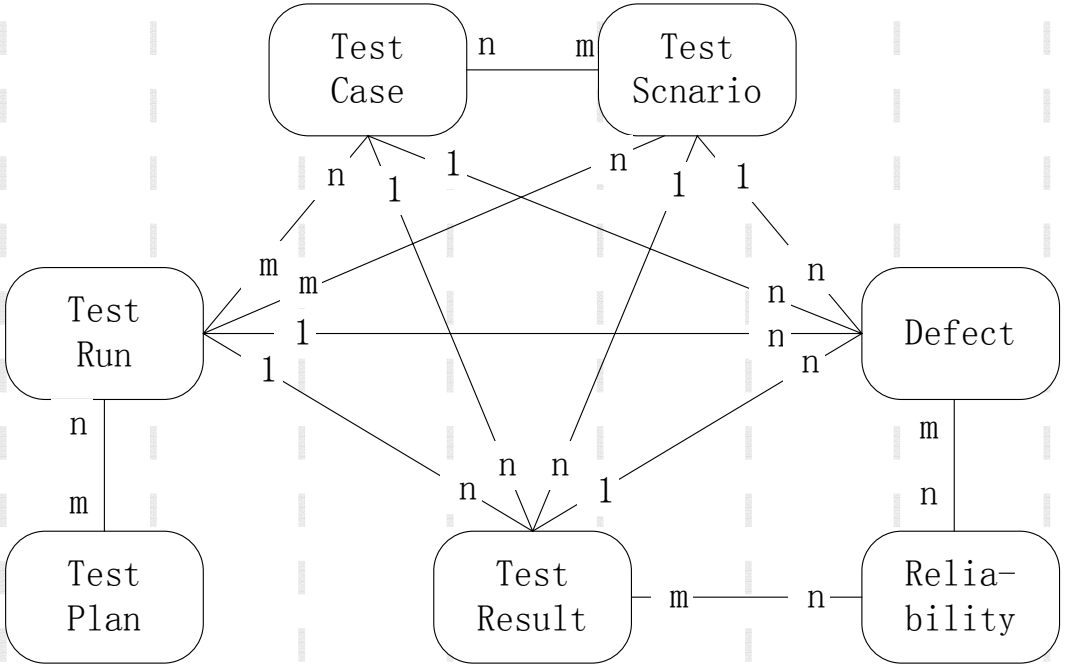
- Multiple loosely coupled test brokers distributed in the Internet environment.
- Each test broker may be dedicated to different V&V tasks or target domains.
- Enable scalable and flexible collaboration among test participants.
- A broker can flexibly join or quit the collaboration. Collaborations can be established at runtime through negotiation.

DCV&V Contracts

- TSC: Testing Service Contracts – Collaboration between test and service under test
  - Test designers get service specification for test generation.
  - Test executors exercise the test on the service interface.
  - Test evaluators evaluate the service based on test results and defect reports.
- TCC: Test Collaboration Contracts – Collaboration among test participants
  - Protocols for collaborative test design, execution, and evaluation.

Test Collaboration Contract

- Test Design
  - Test Case
  - Test Scenario
- Test Scheduling
  - Test Plan
  - Test Run
- Test Execution
  - Test Result
  - Defect
- Test Evaluation
  - Reliability



Contract-Based Test Generation

- OWL-S specification
  - OWL-S introduces Ontology into service representation to improve the mutual comprehension of the operation semantics.
  - ServiceModel models the service composition as a workflow of processes.
    - A composite process holds a Control Construct
    - An atomic process represents a service with IOPE (Input, Output, Precondition, Effect)

Contract-Based Test Generation

- Test Process generation based on Petri-Net model
  - Petri-Net model has a strong capability to model events and states in a distributed system and to capture sequential, concurrency and event-based control.
  - OWL-S processes are mapped to a Petri-Net model.
  - Petri-Net provides powerful support for analyzing and verifying certain properties such like reachability, liveness, and deadlocks.
  - Based on the Petri-Net topology, test processes are generated to cover various execution paths.

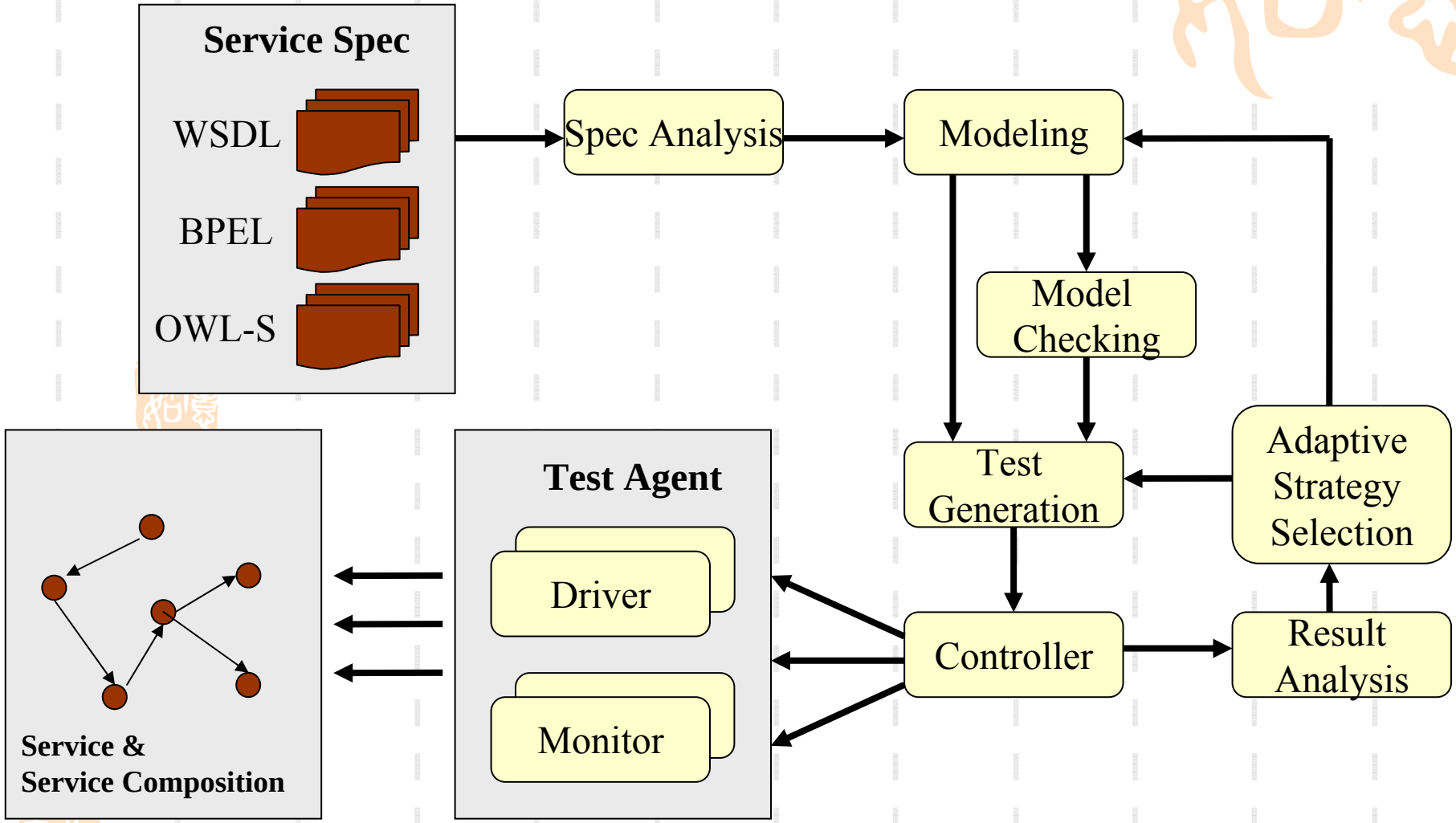
Contract-Based Test Generation

- Constraint-guided test generation
  - WSDL data constraints based on XML schema
    - E.g. {length = ”5”, pattern = ”[A − Z]{2}[0 − 9]{3}”}.
  - OWL-S preconditions

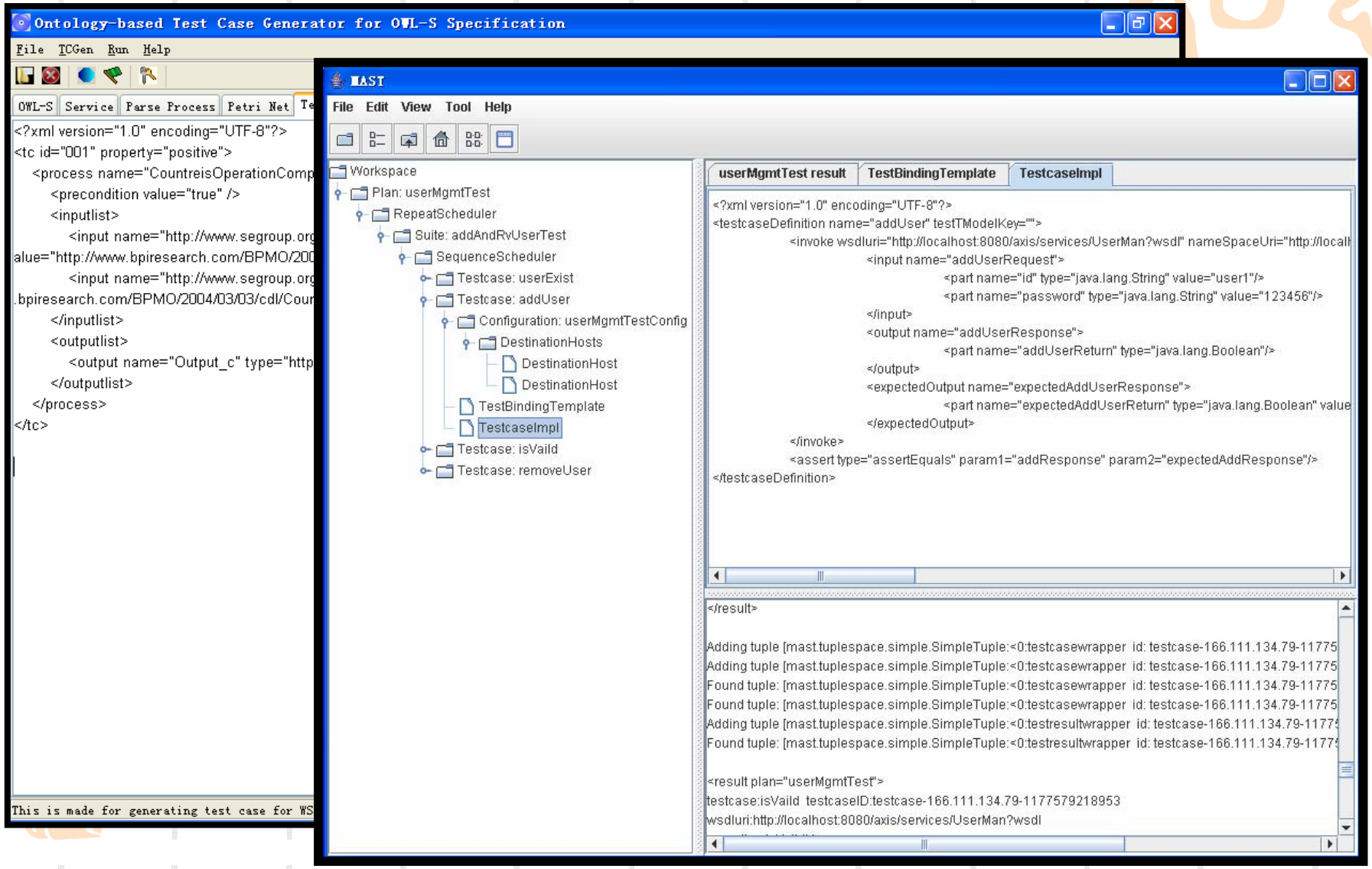
```
<process:inCondition>
    <expr:KIF-Condition>
        <expr:expressionBody>
            (= (?book_room_ID ?select_room_ID))
        </expr:expressionBody>
    </expr:KIF-Condition>
</process:inCondition>
```

  - Process constraints

The Prototype System



The Prototype System



Conclusion

- A test broker architecture is proposed and the prototype system is being developed.
  - The test broker enables collaborations among test participants
  - The test broker collaborates with the service broker to enforce runtime service testing and evaluation
  - The test brokers are loosely coupled and decentralized to enhance scalability. They can dynamically establish collaboration.
- The contracts for broker-based test collaboration is analyzed including TCC and TSC.
- The contract-based automatic testing is discussed.

Thank you!

X. Bai, Y. Wang, G. Dai – Tsinghua University, China
W. T. Tsai, Y. Chen – Arizona State University, US

2007.7