

Towards the Automated Selling of Web Services over the Internet

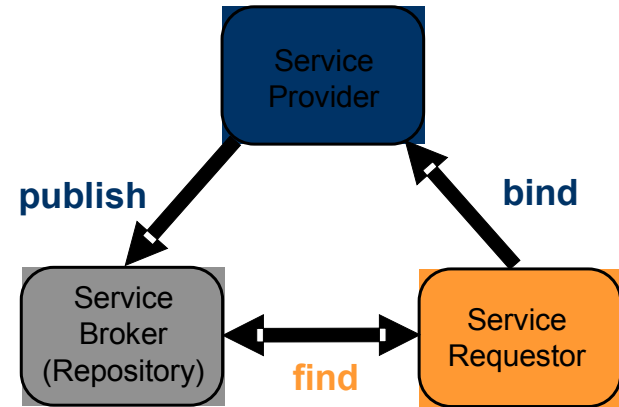
Colin Atkinson
Oliver Hummel
Philipp Bostan

University of Mannheim

AXMEDIS 2006
University of Leeds
December 14th W006

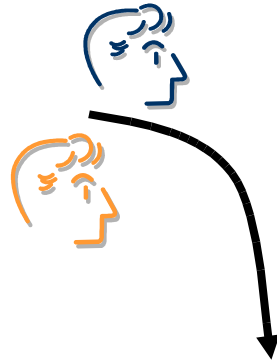
Web Service Brokerage Model

- UDDI is the standard web service brokerage model
 - developers “publish” their services
 - potential consumers “find” services
 - service broker mediates
- however, the first generation of public UDDI repositories did not meet with great success
 - quickly become full of outdated and useless artifacts
- Microsoft, IBM and SAP’s Universal Business Registry was closed down early in 2006



UDDI Usage Scenarios

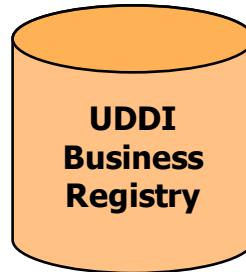
1. Business Analysts, Standard Bodies, Service Providers register descriptions of different kinds of services (WSDL).



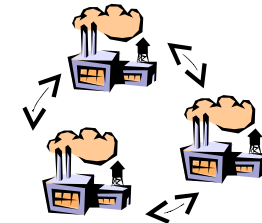
2. "Business" registers which services they support.



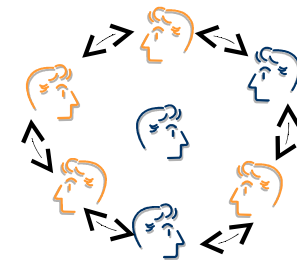
3. Registry assigns a UID or other unique key to each service and business.



5. Service requestors use the fetched data (WSDL) to access the service they need.



4. Service requestors query registry to find the business or services that they want.



Evaluating Fitness for Purpose

- main obstacle to a web service marketplace is reconciling the interests of publishers and consumers
- consumers want to evaluate a service's fitness for purpose but publishers want to control access

Physical artifacts

- can only be used when user is in close proximity
- cannot easily be replicated
- degrade over time
- have physical keys and access controls

Software service

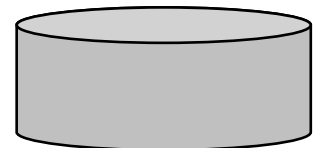
- can be used from anywhere
- can be easily replicated
- do not degrade over time
- have electronic keys and access controls

Service Providers Perspective

- want to control access via trial license keys
 - delivered by e-mail
 - limited the number of allowed accesses
 - valid only for a certain time
- problems and threats
 - identities can easily be copied or faked when a high degree of automation is required
 - re-registration can be automated
- existing solutions
 - give licenses only to customers with contracts
 - distribute license keys by physical mail

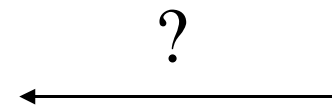
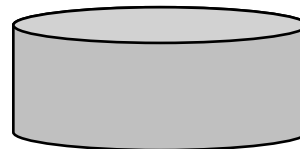


?



Service Consumers Perspective

- wants to establish that a service is fit for purpose
- formal specification and proof techniques unrealistic
 - creating formal specifications can be as complex as programming
 - proving conformance is impossible for all but the smallest services
- the only practical approach is testing
 - potential consumers need to test services before making a purchase decision
 - do not need to know actual results, just the pass or fail evaluation



Solution

- the different interests of the publisher and consumer cannot be reconciled without a third party broker
- the broker must allow the consumer's tests cases to be applied to published services, without
 - the consumer having access to the server
 - the consumer seeing the actual values returned
- the broker must be trusted by both parties
 - the publisher gives the broker full access rights and trusts it not to divulge the license keys
 - the consumer gives the broker complete descriptions of its test cases and trusts it not to divulge them
- service broker must be a trusted testing engine



Username:
Password:
[register](#) (it's free -> [benefit](#))

example queries

- [text »](#)
- [name »](#)
- [function abstraction »](#)
- [object abstraction »](#)
- [object abstraction \(Java/C#\) »](#)
- [web service access »](#)

example constraints

- [class »](#)
- [interface »](#)
- [web service »](#)
- [Java »](#)
- [C# »](#)
- [project »](#)
- [namespace »](#)

your last search

search results

» 1 - 10 of 143 for your query (1594 msec)

1. [wsdl] Sum

- Constructors
- Operations

Relevance: 100.0 % - Author: unknown, unknown license
<http://www.cs.kent.ac.uk/projects/bluej/bluejscrip/soap/delend/Sum-org.wsdl>
(Last Modified: 03/10/2002)

2. [wsdl] Calculate

- Constructors
- Operations

Relevance: 99.0 % - Author: unknown, unknown license
<http://wiki.astrogrid.org/pub/Astrogrid/DocumentStyleExample/CalculateSoapBetter.wsdl>
(Last Modified: 10/06/2003)

3. [wsdl] WrongCalculator

- Constructors
- Operations

Relevance: 99.0 % - Author: unknown, unknown license
<http://authors.aspalliance.com/wisemonk/samples/cal.asm?wsdl>

Example: Number Service (1/3)

- let us assume a service developer has developed a number web service which he would like to publish and earn license fees revenue on
- a license key is obviously needed to stop the service being used by unlicensed users
- but how can people try out the service?
- distributing trial keys is not a good solution
- use a trusted testing broker (TTB)

NumberService

add()

Execute

In:

float

a

3

float

b

4

String

licensekey

?

Out:

float

result

Example: Number Service (2/3)

- publisher provides an unlimited license to the TTB
- all potential consumers can define unlimited test cases (input values and expected result)
- only the correctness of the result is returned not the value

NumberService

add()

Execute

In:

float a

float b

String licensekey *****

Out:

float result

NumberService

add()

Execute

In:

float a

float b

String licensekey *****

Out:

float result

Example: Number Service (3/3)

- the existence of the license key is immaterial to consumers searching for services
- test cases can also be aggregated into tables and executed as a block, as with in the FIT table from Ward Cunningham
- allows correctness of the service to be verified for a potentially large number of test case

a	b	result
3	4	7
5	8	13
1	2	3
4	6	10

NumberService

add()

Execute

In:

float

a

3

float

b

4

Out:

float

result

7

Conclusion

- the fundamental publisher/consumer conflict of interest can (only?) be solved by TTBs
 - establish a relationship of trust with both parties
- definitely applicable to -
 - stateless web services
 - session-driven web services
 - ?
- natural complement to a component/service search engine (e.g. merobase)
 - Extreme harvesting → tests as search filter
- possible TTB business model
 - pay per test (consumer?, publisher)
 - pay per sale (consumer?, publisher)