

Web Services Service Publishing and Discovery (UDDI)

Eetu Mäkelä

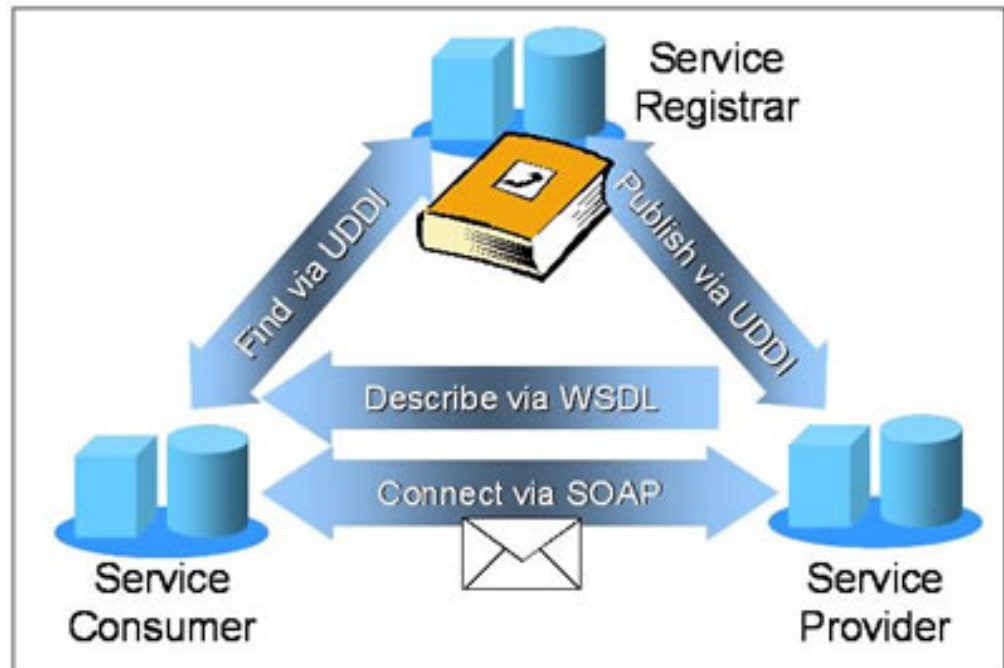
Semantic Computing Research Group
<http://www.cs.helsinki.fi/group/seco/>

Orientation

- Communication between services in general:
 - Find a service via a registry or manually
 - Get a description of the services parameters and interface
 - Locate and/or select an implementation of the service
 - Build a request to the service according to the information exchange protocol and interface and binding descriptions
 - Invoke the service via the information exchange protocol
- Invoking a Web Service:
 - Locate the service manually via UDDI or other registry
 - Get a WSDL interface description from UDDI or via other means
 - Get a WSDL SOAP binding description from UDDI or via other means
 - Build a SOAP request according to the WSDL descriptions
 - Call service via the SOAP binding

Orientation

- In other words, service registries (like UDDI) provide **catalogues** of services, and allow the providers and consumers of these services a place to **locate** each other



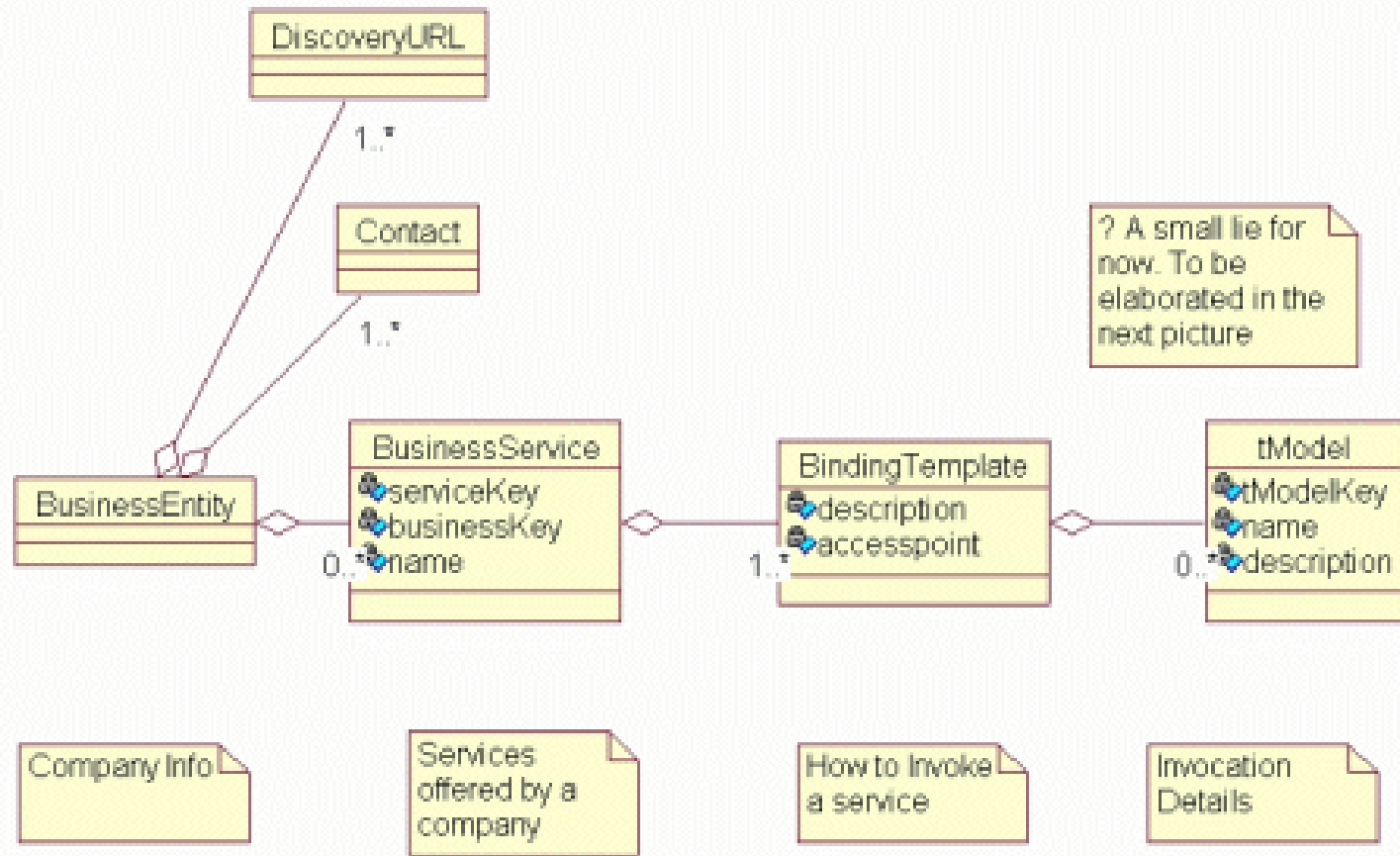
Ideals and Reality in Service Registries

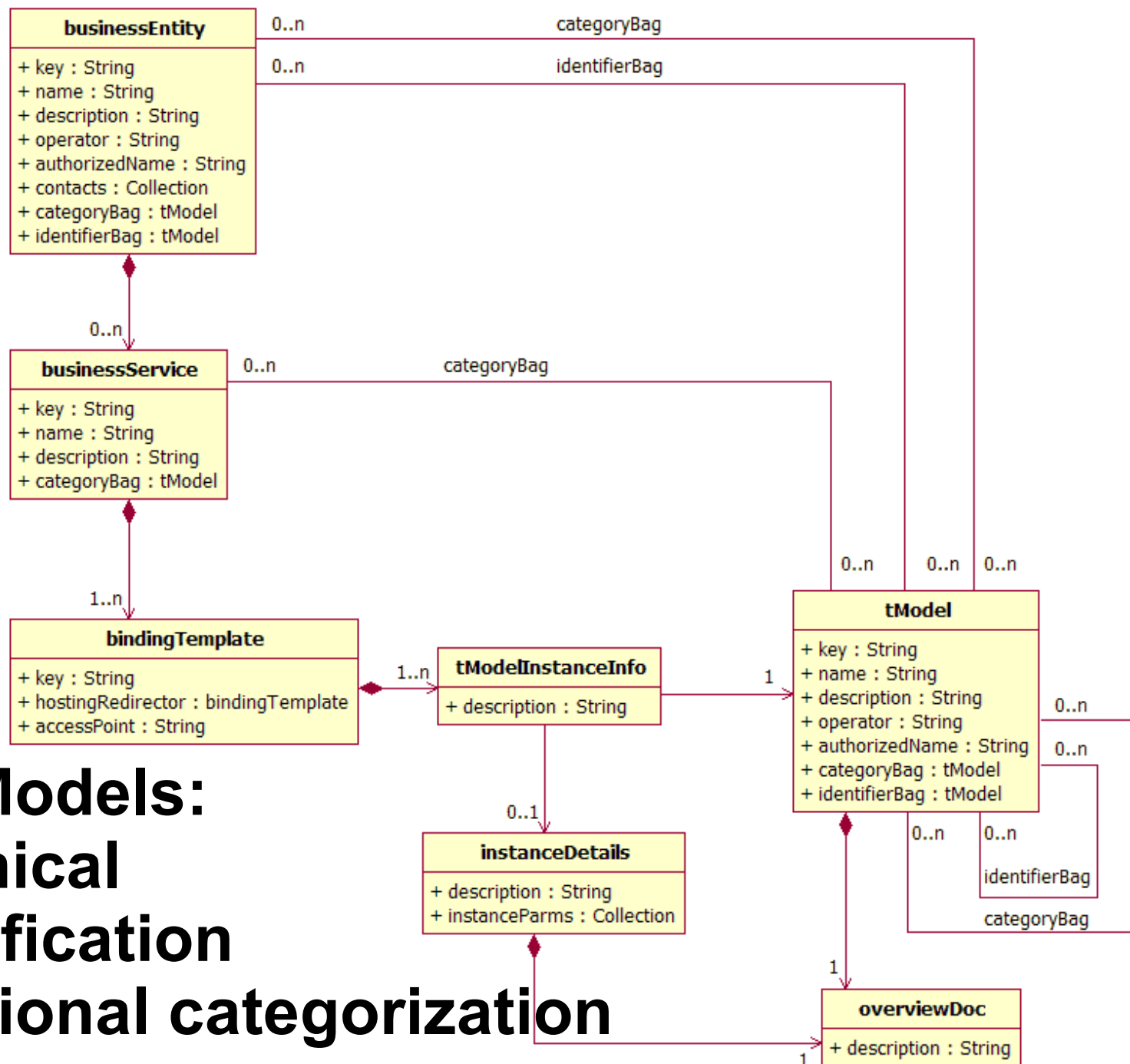
- In the full SOA vision (needed for use cases B2C and C2C), intelligent applications should be able to programmatically and robustly
 - Locate the services they need
 - Know how to map their service need into terms the service implementation understands
 - Combine the services into useful transactions
- In actuality (sufficient for the B2B use cases):
 - Prospective services are located manually via registries and manually evaluated for suitability. Automatic discovery is only useful if multiple service providers implement a known rigid, agreed-upon interface definition
 - Applications are bound by hand to the exact implementation semantics of the services they are using
 - Translations between service semantics need to be done by hand

UDDI, Original Vision and Reality

- Vision: “Universal Description, Discovery and Integration”, with advanced functionality
- Reality:
 - Keyword-based search
 - Few public registries:
 - Company contact information
 - Company/service yellow pages, with a very general categorization, usable only for manual search
 - Mostly used as private binding information directories
 - Service implementation search by known WSDL interface definition
 - Technically very extendable with regard to content (but complex)

UDDI Registry Layout





UDDI tModels:

1) technical
specification

2) functional categorization

UDDI Demo

- <http://soapclient.com/uddisearch.html>
- <http://www.xmethods.com/>

“And How do I Actually Use It?”, You Ask

- JAXR: Java API for XML Registries

tModel data structure	JAXR Concept
tModelKey	<code>Concept.getKey()</code>
<code>authorizedName</code>	<code>Concept.getSlot()</code>
<code>operator</code>	<code>Concept.getSlot()</code>
name	<code>Concept.getName()</code>
<code>description</code>	<code>Concept.getDescription()</code>
<code>overviewDoc</code>	<code>Concept.getExternalLinks()</code>
<code>identifierBag</code>	<code>Concept.getExternalIdentifiers()</code>
<code>categoryBag</code>	<code>Concept.getClassifications()</code>

Returning to that Vision

- To programmatically locate and utilize services, much more robust service descriptions are needed
 - Semantic descriptions of what the service does (travel planning vs. train schedule planning vs. air route planning)
 - Semantic descriptions of what the service requires (Firstname Lastname parameters vs. Lastname, Firstname, Title, First call order service to get invoice, then call bank service to pay it vs. call combined order & pay service)
 - Semantic descriptions of additional constraints and guidelines, such as cost, privacy, etc. (Usage of this service costs 5 EUR, usage of that service 5 USD)
 - Systems and standards that are able to automatically utilize this complex information
 - Hot research topic

Thank You

- Any questions or comments?