

Web Services Service Publishing and Discovery (UDDI)

Eetu Mäkelä

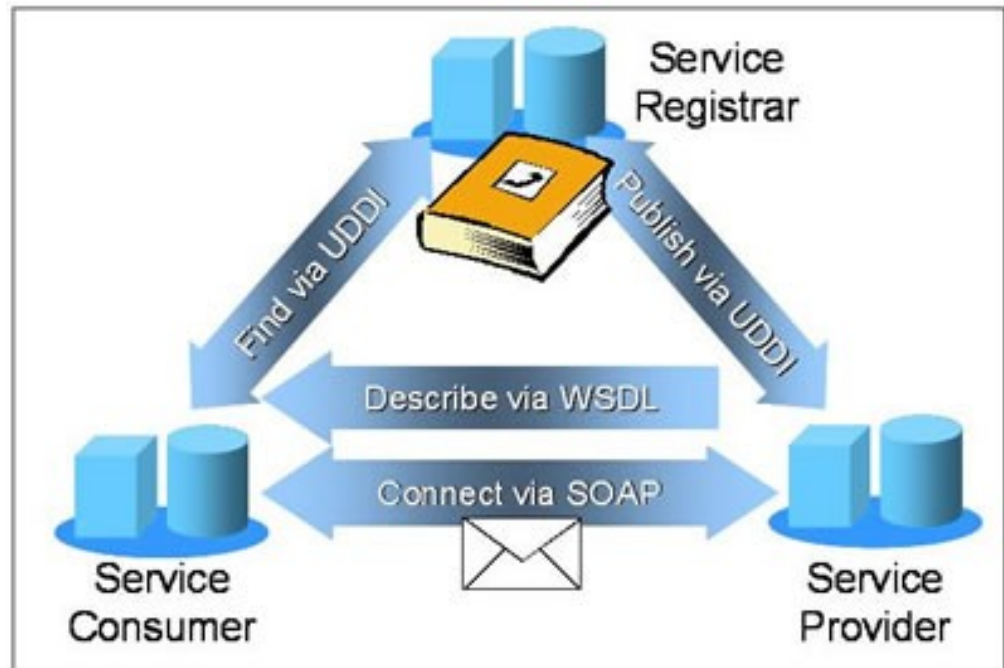
Semantic Computing Research Group
<http://www.cs.helsinki.fi/group/seco/>

Orientation

- Communication between services in general:
 - 1) Find a service via a registry or manually
 - 2) Get a description of the services parameters and interface
 - 3) Locate and/or select an implementation of the service
 - 4) Build a request to the service according to the information exchange protocol and interface and binding descriptions
 - 5) Invoke the service via the information exchange protocol
- Invoking a Web Service:
 - 1) Locate the service manually via UDDI or other registry
 - 2) Get a WSDL interface description from UDDI or via other means
 - 3) Get a WSDL SOAP binding description from UDDI or via other means
 - 4) Build a SOAP request according to the WSDL descriptions
 - 5) 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



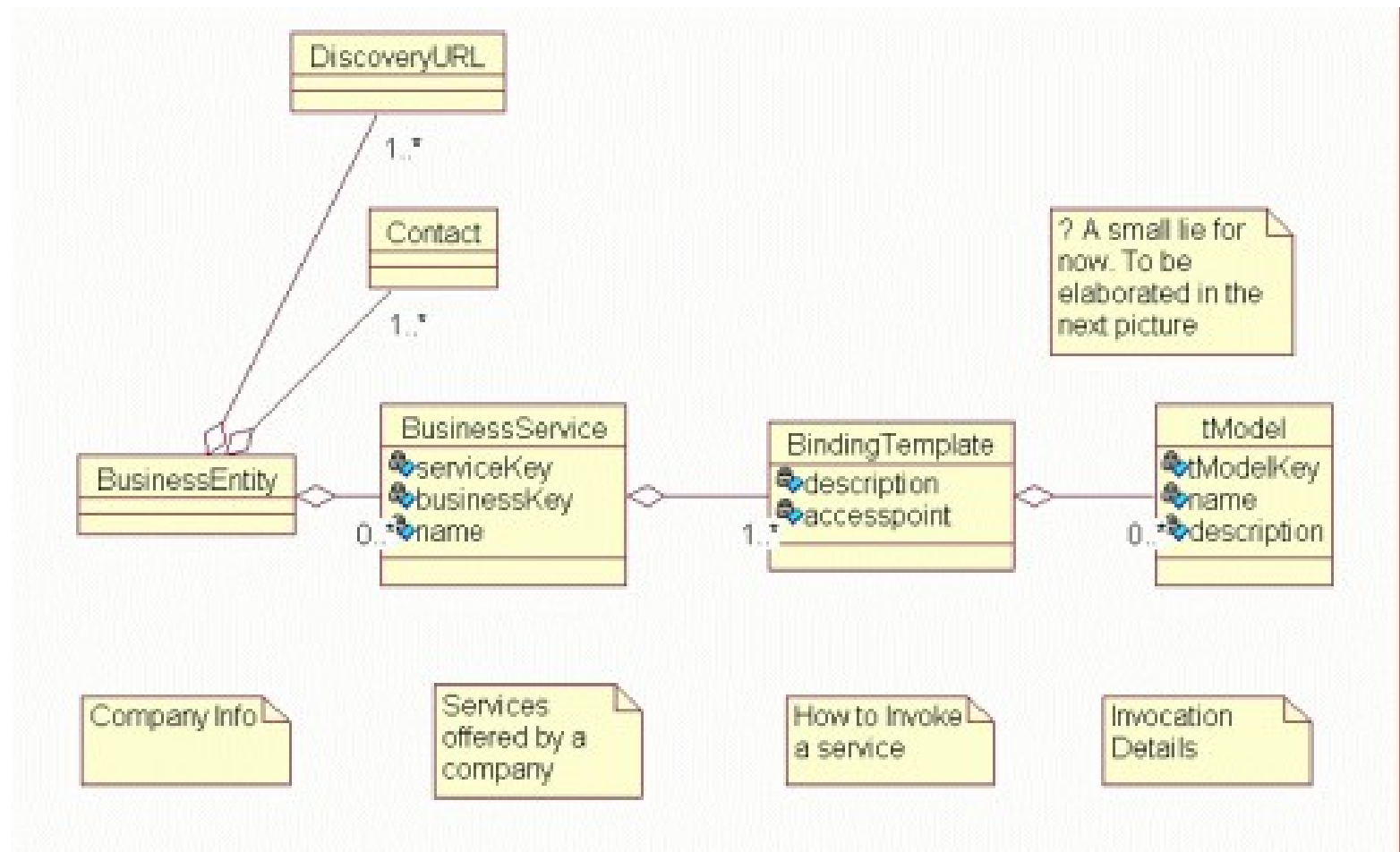
Ideals and Reality in Service Registries

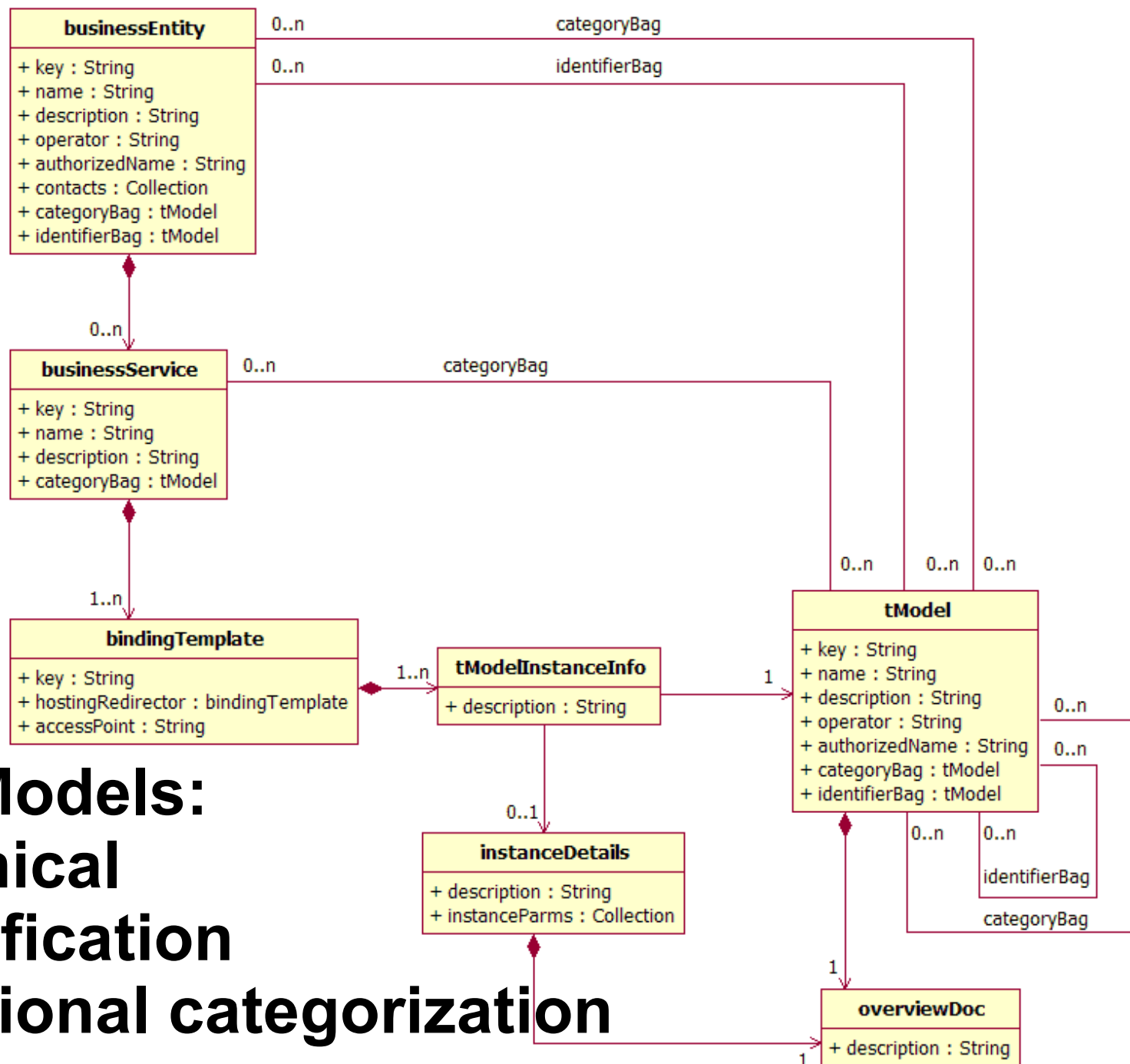
- In the full SOA vision (needed for use cases B2C and C2C), intelligent applications should be able to programmatically and robustly
 - Locate the services they need
 - Know how to map their service need into terms the service implementation understands
 - Combine the services into useful transactions
- In actuality (sufficient for the two 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>
authorizedName	<code>Concept.getSlot()</code>
operator	<code>Concept.getSlot()</code>
name	<code>Concept.getName()</code>
description	<code>Concept.getDescription()</code>
overviewDoc	<code>Concept.getExternalLinks()</code>
identifierBag	<code>Concept.getExternalIdentifiers()</code>
categoryBag	<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
 - Semantic descriptions of what the service requires
 - Semantic descriptions of additional constraints and guidelines, such as cost, privacy, etc.
 - Systems and standards that are able to automatically utilize this complex information
 - Hot research topic

Thank You

- Any questions or comments?