Combining SAWSDL, OWL-DL and UDDI for Semantically Enhanced Web Service Discovery

Dimitrios Kourtesis, Iraklis Paraskakis

SEERC - South East European Research Centre, Greece
Research centre of the University of Sheffield and CITY College
http://www.seerc.org

Outline

1. Background and motivation
2. Describing service characteristics in Semantic Registry matchmaking
3. Selection capabilities
4. Overview of architecture and functionality
5. Summary

1. Background and motivation
2. Describing service characteristics in Semantic Registry matchmaking
3. Persistent capabilities
4. Overview of architecture and functionality
5. Summary

Background

- **Enterprise interoperability** is a major challenge
  - **Business agility** calls for integrating heterogeneous business applications
- **SOA** and **Web service technologies** can help
  - SOA helps align the IT and business perspectives
  - Web Service standards promote (transport-level) interoperability
- Business process integration in service-oriented style:
  - Assembling **reusable services** from **heterogeneous applications** into new service **compositions**
  - A suitable **interoperable service** must be **discovered** for each task/step in the envisaged process
- Locating interoperable services can become very complex
  - We need automation in service discovery!

Universal Description, Discovery, and Integration

- To automate discovery, service characteristics must be described:
  - in sufficient detail
  - in a structured, uniform way
- The UDDI specification by OASIS was a first step to this direction by defining:
  - an XML-based data model for describing services and providers
  - a Web service-based API for publication & discovery in a registry
- A core WS technology standard with universal endorsement by business software vendors
  - IBM WebSphere UDDI Registry, Oracle Service Registry, SAP Enterprise Services Registry, Microsoft Windows Server 2003 Enterprise UDDI Services, ...

Description and discovery facilities of UDDI registries

- Services in UDDI can be described by:
  - references to externally maintained technical specifications (e.g. a WSDL document)
  - references to internally represented classification schemes (pre-existing or created ad-hoc)
  - free-text information (in service-name and service-description fields)
- Services in UDDI can be discovered by:
  - searching for a matching categorisation value
    - i.e. searching for declared conformance to some technical specification or to some specific entity in a classification scheme
  - searching for a matching keyword

Shortcomings of UDDI

- Major shortcoming - no way to impose uniformity, level of granularity and standard semantics in:
  - the externally maintained technical specifications
  - the internally-represented classification schemes
- As a result:
  - Service descriptions are machine-processable but lack the machine-understandable semantics for fully automated processing
  - Fine-grained matchmaking (e.g. to assert data-level interoperability) is impossible
- Developers typically need to inspect service specifications **manually** to evaluate interoperability

Motivation for semantic enhancements to UDDI

- Semantic enhancements to UDDI should enable:
  - **formal and unambiguous** representations of service properties
  - **fine-grained** matchmaking among service advertisements and service requests via **logic-based reasoning**
  - **fully-automated** service discovery
- …while leveraging the extensive industrial adoption of the UDDI specification

Semantic enhancements to UDDI

- Semantic enhancements to UDDI encompass two requirements:
  1. A semantic formalism for representing service characteristics
  2. A technical infrastructure complementing UDDI and offering semantically-enhanced publication and discovery functions
- The rest of this presentation focuses on how these semantic enhancements have been realised in the context of project FUSION

Research project FUSION

- FP6 -funded STREP project, 12 industrial and academic partners, coordinated by SAP
- Focus:
  - promoting collaboration within and among enterprises by developing technologies for the semantic fusion of heterogeneous business applications
- Research outcomes:
  - a Semantic EAI reference framework, a supporting methodology, and a set of tools for realising Enterprise Application Integration (EAI) through Semantic Web Services
- Semantically-enhanced service discovery based on widely accepted standards is an essential requirement for the approach that FUSION puts forward

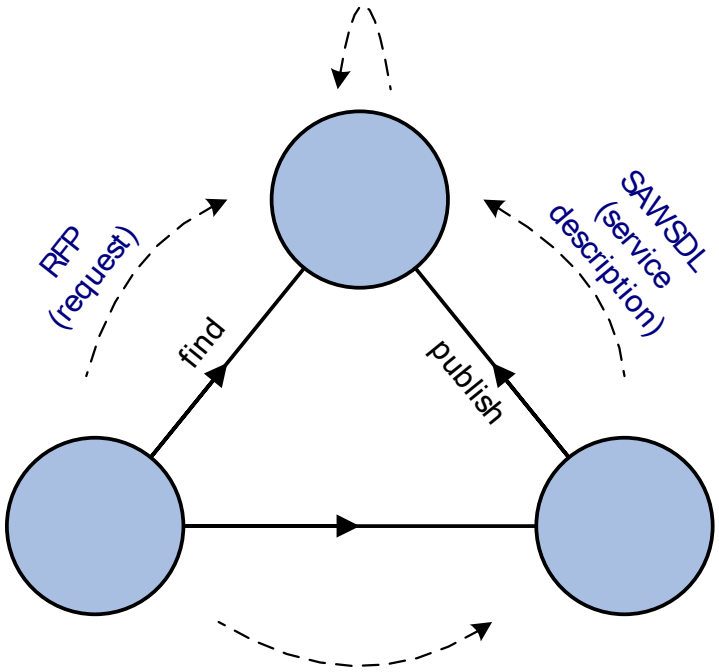
1. Background and motivation
2. Describing service characteristics in Semantic Registry matchmaking
3. Describing service capabilities
4. Overview of architecture and functionality
5. Summary

The FUSION Ontology

- The FUSION Ontology is the semantic model used for representing and reasoning upon service characteristics
- The FUSION Ontology allows modelling:
  - the **data structures** a Web service exchanges through input and output messages
  - the **functionality categorisation** of a service with regard to a classification taxonomy
  - the **behaviour** it may expose within a complex and stateful process execution
  - the **collaborative business processes** that a service may belong to

Semantic representation of service characteristics (1/2)

- Service characteristics are represented using **Functional Profiles**:
  - to describe services advertised by **providers**
  - to describe services sought by **requestors**
  - to perform matchmaking inside the **registry**
- Two types of FPs:
  - Advertisement Functional Profiles
  - Request Functional Profiles



Semantic representation of service characteristics (2/2)

- Functional Profiles are light-weight means of capturing functional and non-functional characteristics:
  - service inputs, service outputs, and service categorisation
- FPs are expressed as OWL concepts in the FUSION Ontology and are associated with other FUSION Ontology concepts through OWL object properties:
  - **hasCategory**: pointing to a concept representing the category of a service
  - **hasInput**: pointing to a concept representing the set of data parameters that a service expects to receive when invoked
  - **hasOutput**: pointing to a concept representing the set of data parameters that a service will produce (upon success or failure)

Matchmaking with Functional Profiles

- Semantic matchmaking in FUSION aims at asserting **data-level interoperability** among service providers and service consumers:
  - **Input compatibility**: the service requestor intends to provide the input data that the provider expects, or even more
  - **Output compatibility**: the service provider will produce the output data that the requestor demands, or even more
- Categorisation-level matchmaking is an optional utility for quickly filtering-out irrelevant services
  - **Category compatibility**: the classification value of a service is more specialised or equivalent to that specified by the requestor

Matchmaking criteria

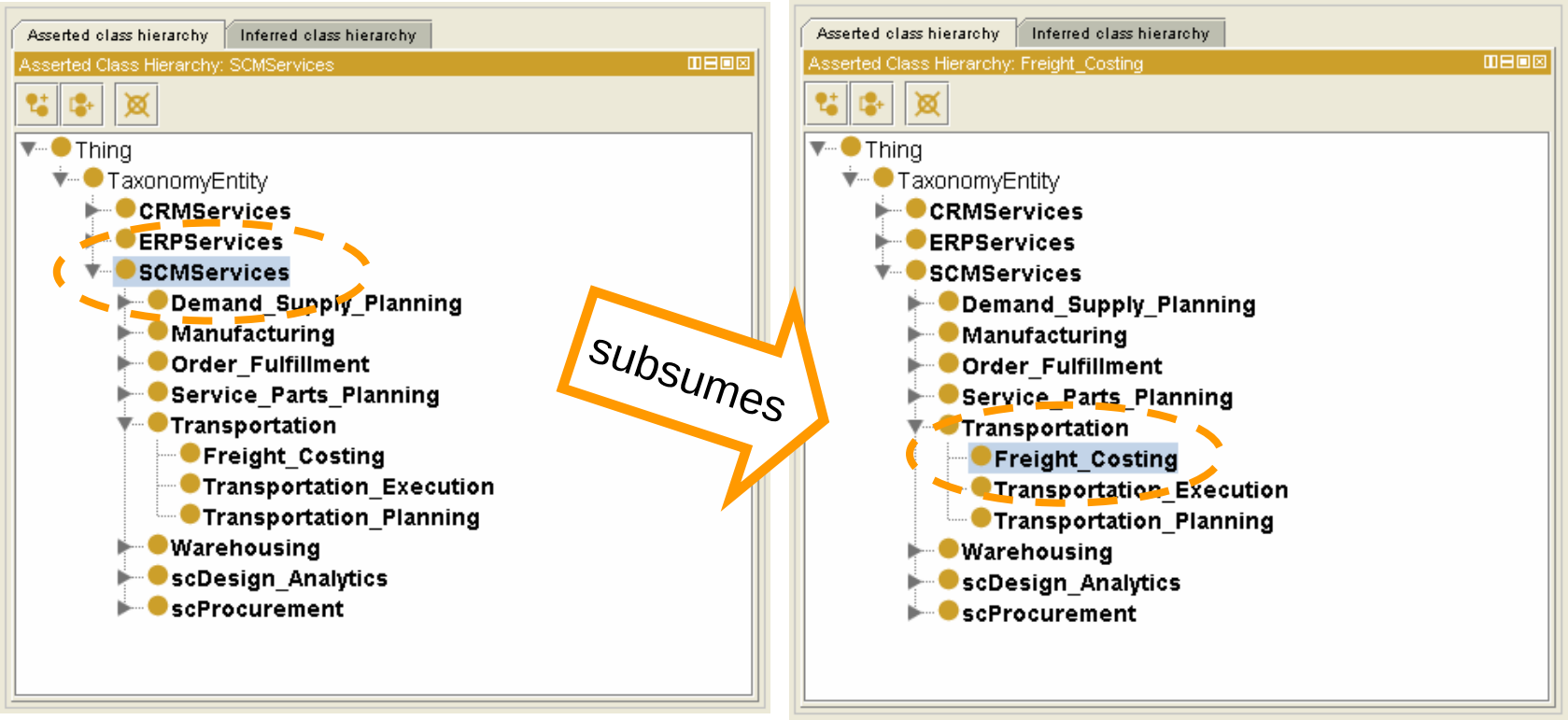
- In DL terms, an advertisement matches a request if:
  - hasCategory concept of RFP **subsumes** that of the AFP
  - hasInput data-set of RFP **is subsumed by** that of the AFP
  - hasOutput data-set of RFP **subsumes** that of the AFP
- Final matchmaking decision:
  - combination of three independent subsumption checks
  - depending also on service Message Exchange Pattern
- Main advantage of describing services with OWL-DL:
  - Existing DL reasoners can be used for performing optimised subsumption checking

1. Background and motivation
2. Describing service characteristics in Semantic Registry matchmaking
3. Design capabilities
4. Overview of architecture and functionality
5. Summary

Matchmaking capabilities in practical terms

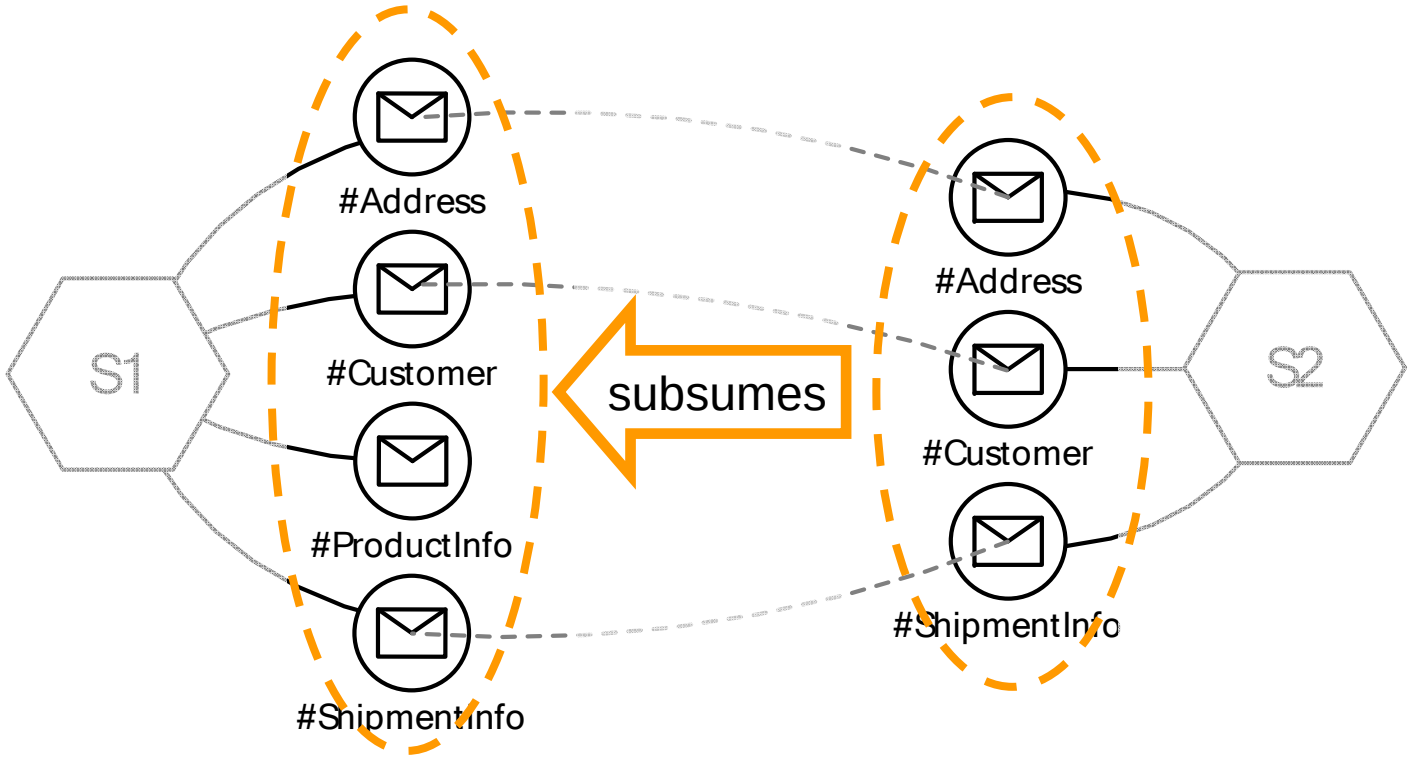
- From the perspective of an end-user, relevance among a request and an advertisement is evaluated at 3 levels:
  - Equivalence of functionality categorisations
  - Compatibility of elements in I/O messages
  - Compatibility of schema structures in message elements (fine-grained matching)

Example: evaluation of category-level matching



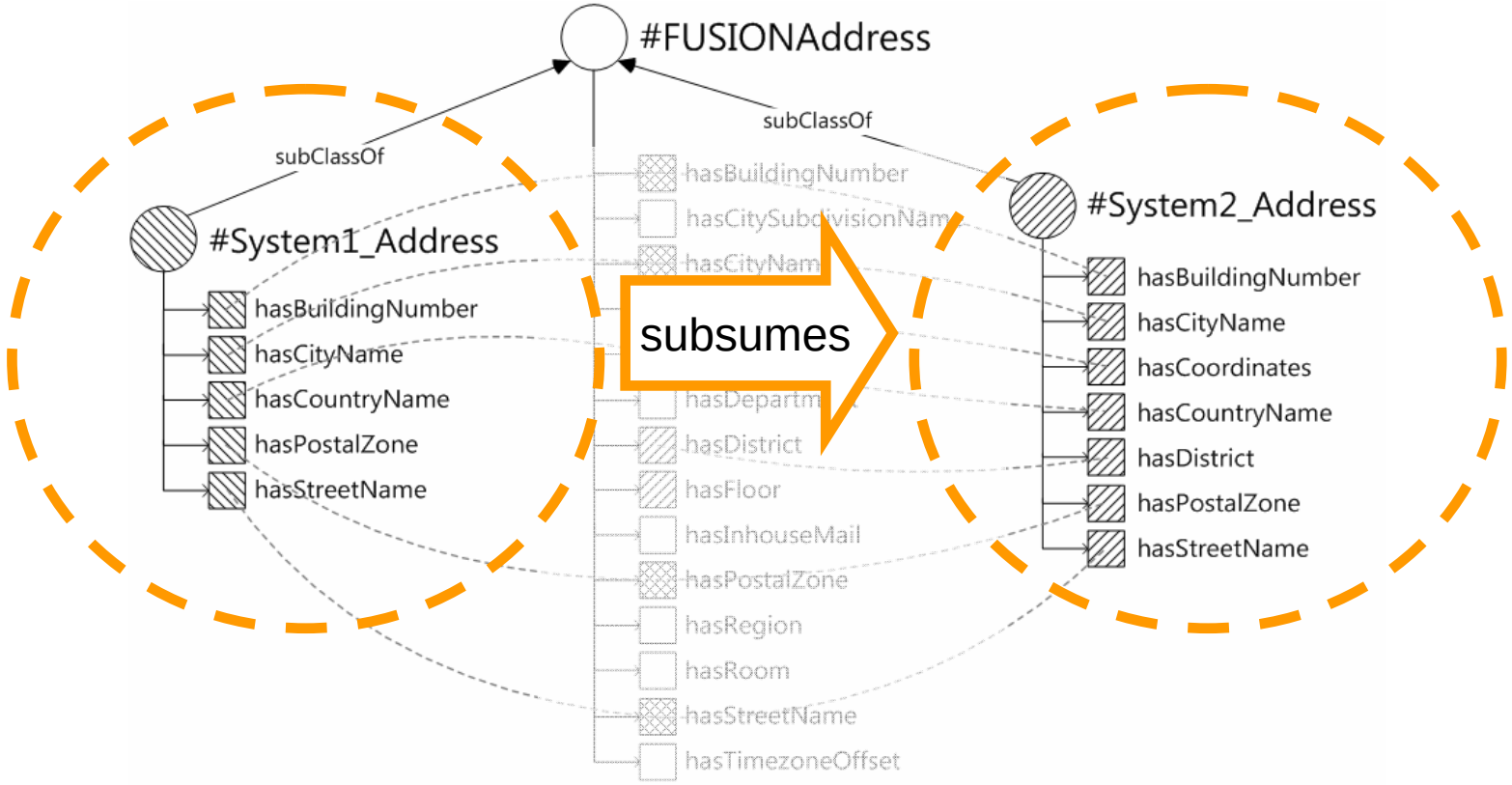
- If SCM Services was requested: **Positive match!**
- If Freight Costing was requested : **Non-match!**

Example: evaluation of message-level matching



- If System1 provided input to System2: **Positive match!**
- If System2 provided input to System1: **Interoperability problem!**

Example: evaluation of schema-level matching



- If System2 provided input to System1: **Positive match!**
- If System1 provided input to System2: **Interoperability problem!**

1 Introduction and motivation

2 Describing service characteristics in Semantic Registry matchmaking

3 Persistent capabilities

4 Overview of architecture and functionality

5 Summary

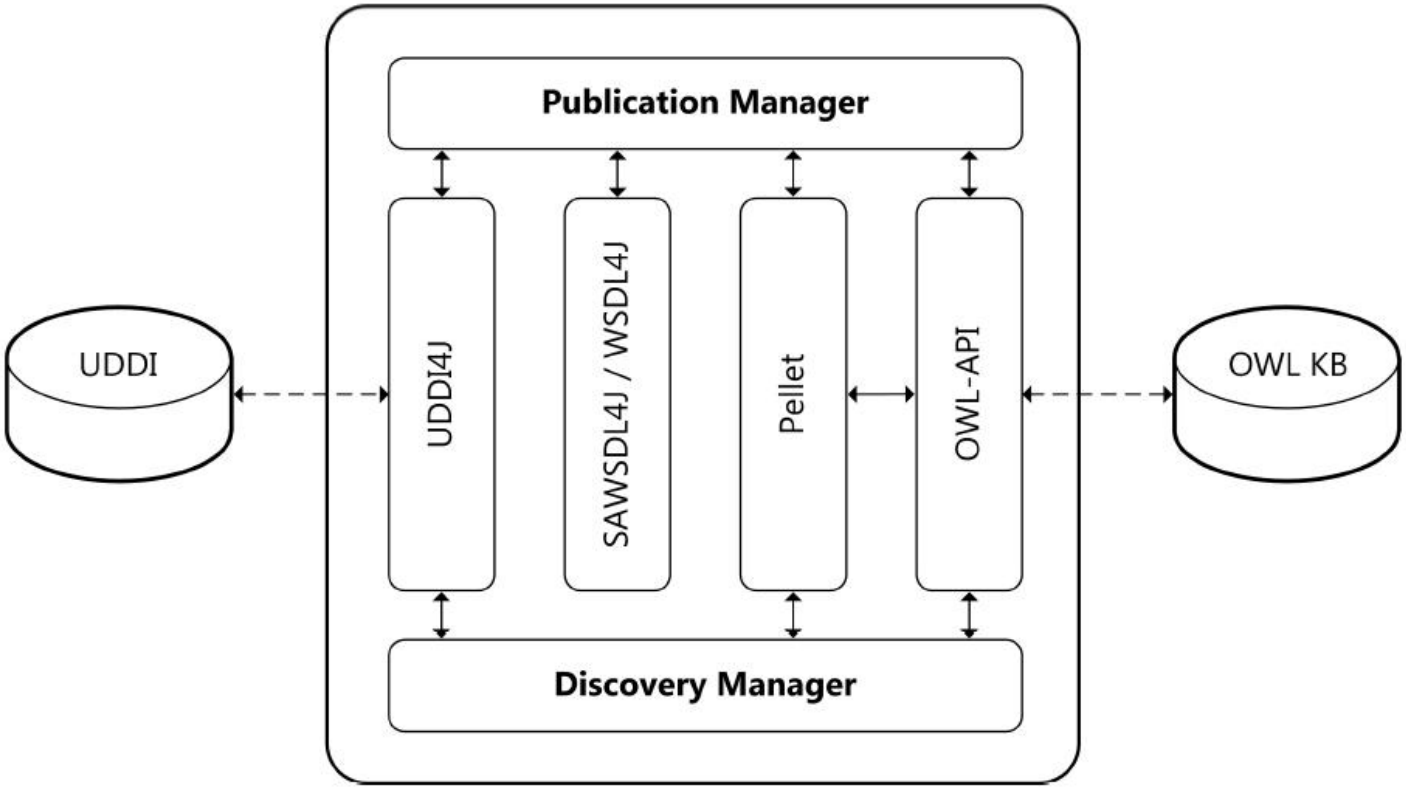
Design considerations

- Key design considerations for the Semantic Registry:
  - **No modifications to the UDDI server** implementation or to UDDI API
  - **Semantically-enabled processing modules** are layered on top of UDDI (performing **OWL ontology processing** and **DL reasoning**)
- The core UDDI server module in the system remains **semantics-agnostic**
- Implementers can use **any server implementation** of the UDDI v2 or v3 specification (no vendor lock-in)
- Syntactic and semantic search can be even used interchangeably

Core functionality

- Publication operations:
  - Adding a service/provider record
  - Modifying a service/provider record
  - Removing a service/provider record
- Discovery operations:
  - Retrieving a service/provider record via its key
  - Discovering a set of service/provider records matching a keyword
  - Discovering a set of service records matching a Request Functional Profile

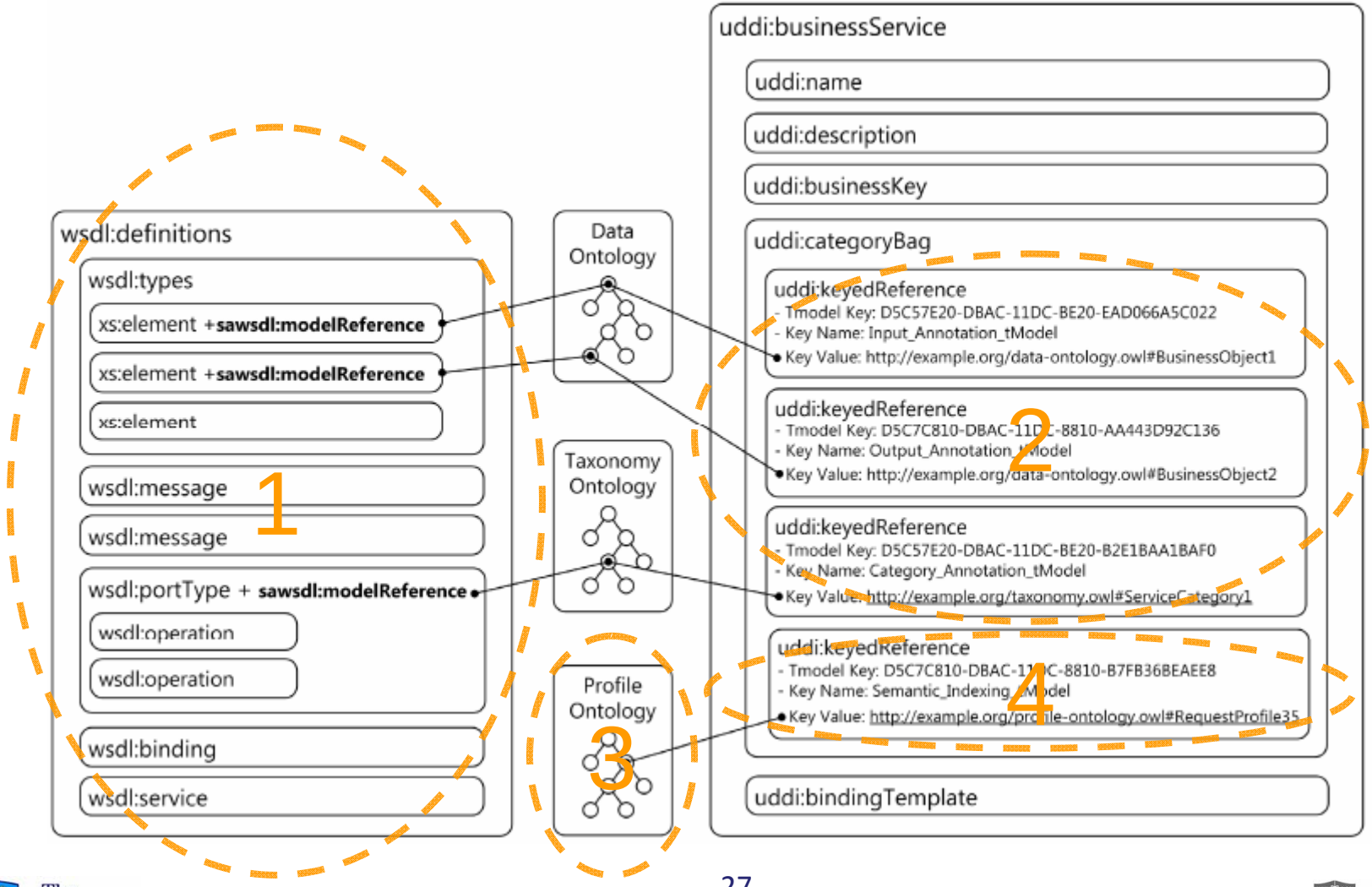
Semantic Registry architecture overview



Service publication process

1. Parsing of SAWSDL document
   - extraction of semantic annotations (I/O & category)
2. Construction of UDDI advertisement
   - information mapped to UDDI structures
3. Generation of Functional Profile and publication-time matchmaking (eager)
   - AFP constructed and stored in OWL KB
   - subsumption hierarchy computed for KB and RFPs are matched against AFPs
4. Indexing of semantic matching results
   - update of UDDI structures

SAWSDL to UDDI mapping



Service discovery process

- Collaborative Business Process templates contain references to Request Functional Profiles
- The URI of each RFP is provided as query to the registry
- If the RFP is defined in the standard EAI ontology:
  - The registry looks for UDDI advertisements already indexed with a reference to the specified RFP
  - The uddi:businessService entities that correspond to matching advertisements are retrieved and returned
- Else:
  - Ontology loaded to reasoner for discovery-time matchmaking
  - AFPs matching the RFP are identified
  - The uddi:businessService entities of services corresponding to these AFPs are retrieved and returned

Implementation details

- Developed as a Web Application offering a set of WSs
  1. PublicationManager WS
  2. DiscoveryManager WS
  3. AdministrationManager WS
- Packaged and deployed on any standard Java Servlet container (tested with Apache Tomcat 5.5)
- Dependencies:
  - UDDI v2 or v3 compliant server (tested with Apache jUDDI)
  - SAWSDL4J and WSDL4J (for parsing SAWSDL documents)
  - UDDI4J (for communicating with UDDI)
  - OWL API (for manipulating OWL constructs)
  - Pellet (for Description Logics reasoning)

1. Introduction and motivation
2. Describing service characteristics in Semantic Registry matchmaking
3. Persistent capabilities
4. Overview of architecture and functionality
5. Summary

Summary

- Automated service discovery is a key challenge for integration within and between enterprises
- Semantics are key for enabling automated service discovery, and overcoming the limitations of UDDI
- The FUSION Semantic Registry:
  - A semantically-enhanced Web service registry building on UDDI, SAWSDL and OWL
  - Augments and enhances the discovery facilities that a typical UDDI registry can offer
  - Layered on top of UDDI without imposing changes to the UDDI registry's implementation or API
  - Performs fine-grained matchmaking with an emphasis in asserting data-level interoperability of services
  - Developed by SEERC in the context of research project FUSION and released as open source software

Future work

- Expanding the registry's matchmaking capabilities:
  - Matchmaking based on behavioural semantics (preconditions and effects)
  - Matchmaking based on non-functional properties (compliance to policies, SLAs, ...)
- Expanding the scope of the registry:
  - Addition of repository functions for handling semantic metadata
  - Certification of services through registry-based testing and verification
- Focus on realising a semantically-enhanced registry and repository to support service lifecycle governance

SOUTH-EAST EUROPEAN RESEARCH CENTRE

Thank you!

More information at:

http://www.seerc.org/fusion/semanticregistry

The University Of Sheffield.

CITY
Affiliated Institution of the
UNIVERSITY OF SHEFFIELD