UNIVERSITY
OF TRENTO - Italy

Web Service Discovery with Implicit QoS Filtering

Natallia Kokash,

DIT - University of Trento

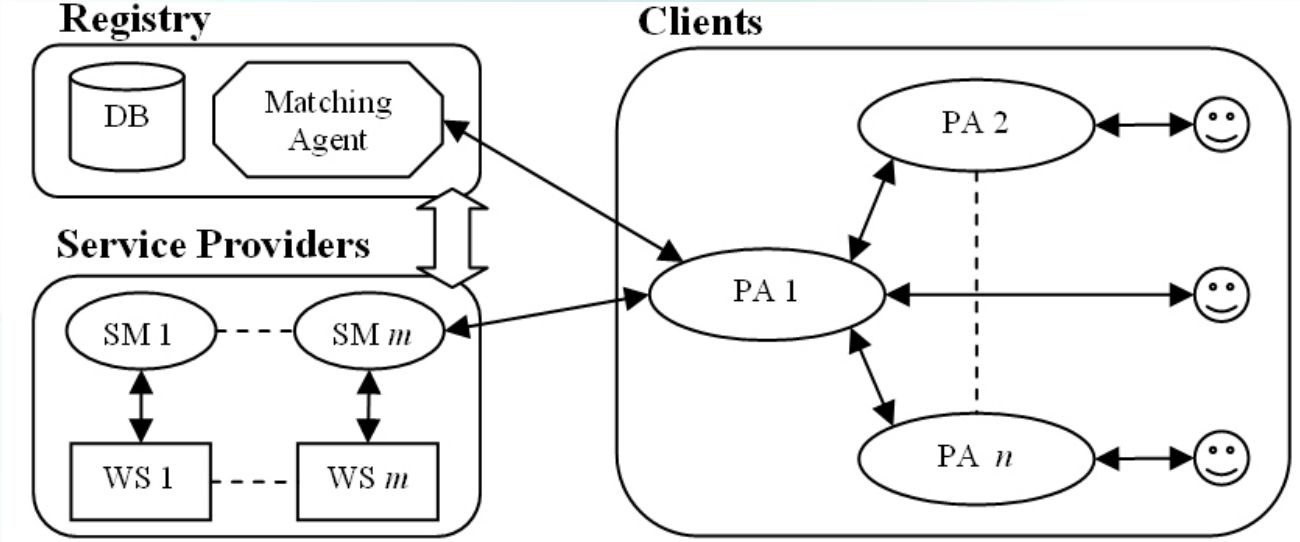
Introduction

- **Web Service (WS) Discovery Problem**
- **WS Discovery Framework**
  - **Web Services Matching**
    - **Related work**
    - **Matching Algorithm**
  - **Quality of Service (QoS) Issues**
  - **Framework Implementation**
- **Future Work**
- **References**
- **Questions**

Web Service Discovery Problem

- **Web Service Discovery Problem**
  - **Identification of existing services that can be used by new web applications**
    - **Locate a service that satisfies user's goal**
    - **Provide QoS characteristics**
    - **Domain and user specific requirements**
  - ***Automated discovery* – search and result evaluation is performed by agent**
- **Limitations of existing solutions (UDDI, ebXML registries)**
  - **Inefficient matching**
  - **No QoS-aware requests**

Framework



- **Each user has a Personal Agent (PA)** for discovery and invocation monitoring of WSs
- **Web Services can have Service Mediators (SMs)** for invocation monitoring
- **Related work**
  - [Maximilien02]

WS Matching. Related work

- **Text document retrieval (exploiting relative frequencies of words and word combinations)**
  - **Vector-Space Model**
  - **Latent Semantic Indexing (LSI)**
  - **Probabilistic Models (e.g., Probabilistic LSI)**
- **Semi-structured documents (HTML, XML)**
- **Schema matching**
  - **User-assisted iterative matching**
  - **Soundex (e.g., txtToPdf = txt2Pdf)**
- **Software component matching**

- **Application to Web Services:**
  - **Matching of specifications (WSDL files)**
    - **[Stroulia05]**
    - **[Dong04]**
  - **Retrieval from registries**
    - **LSI on UDDI descriptions [Sajjanhar04]**

WS Matching

Vector Space Model

$$D = \{d_1, ..., d_n\}$$

$$\forall d_i, i = \overline{1, n}, x_i = \{x_{i1}, ..., x_{im}\}$$

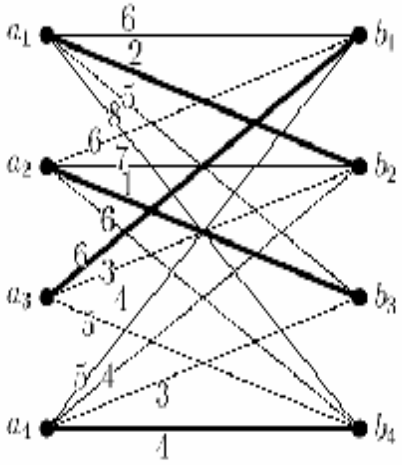
$$TF_{ij} = \frac{n_{ij}}{|d_i|}$$

$$IDF_j = \log(\frac{n}{n_j})$$

$$x_{ij} = TF_{ij} \cdot IDF_j$$

$$cos(x, x') = \frac{x^T x'}{\sqrt{x^T x}\sqrt{x'^T x'}}$$

Maximum weight bipartite matching (Hungarian Algorithm)



WS Matching

- **Part names**
  1. **Extract semantic data from the element description in WSDL file (part names, names of complex types, documentation, etc.)**
  2. **Extract meaningful words from the element description obtained on step 1. Perform word stemming.**
  3. **Enrich the extracted set of words with the semantically close concepts from the WordNet database or user vocabulary to enable synonym recognition.**
  4. **Assign weights to each word using TF-IDF measure.**
  5. **Define the similarity of two elements using cosine coefficient.**
- **Messages**
  1. **For any pair of part names define similarity like described above. Compose a matrix from these weights.**
  2. **Apply type filter (if we want to take into account data types):**
     - **w_{ij} = 1 if it is impossible to convert a type of parameter *i* into a type of parameter *j;***
     - **w_{ij} = 0 otherwise.**
  3. **Apply the algorithm for Maximum Weight Bipartite Matching problem (e.g., Hungarian method) to find the best assignment of part names.**

QoS Issues

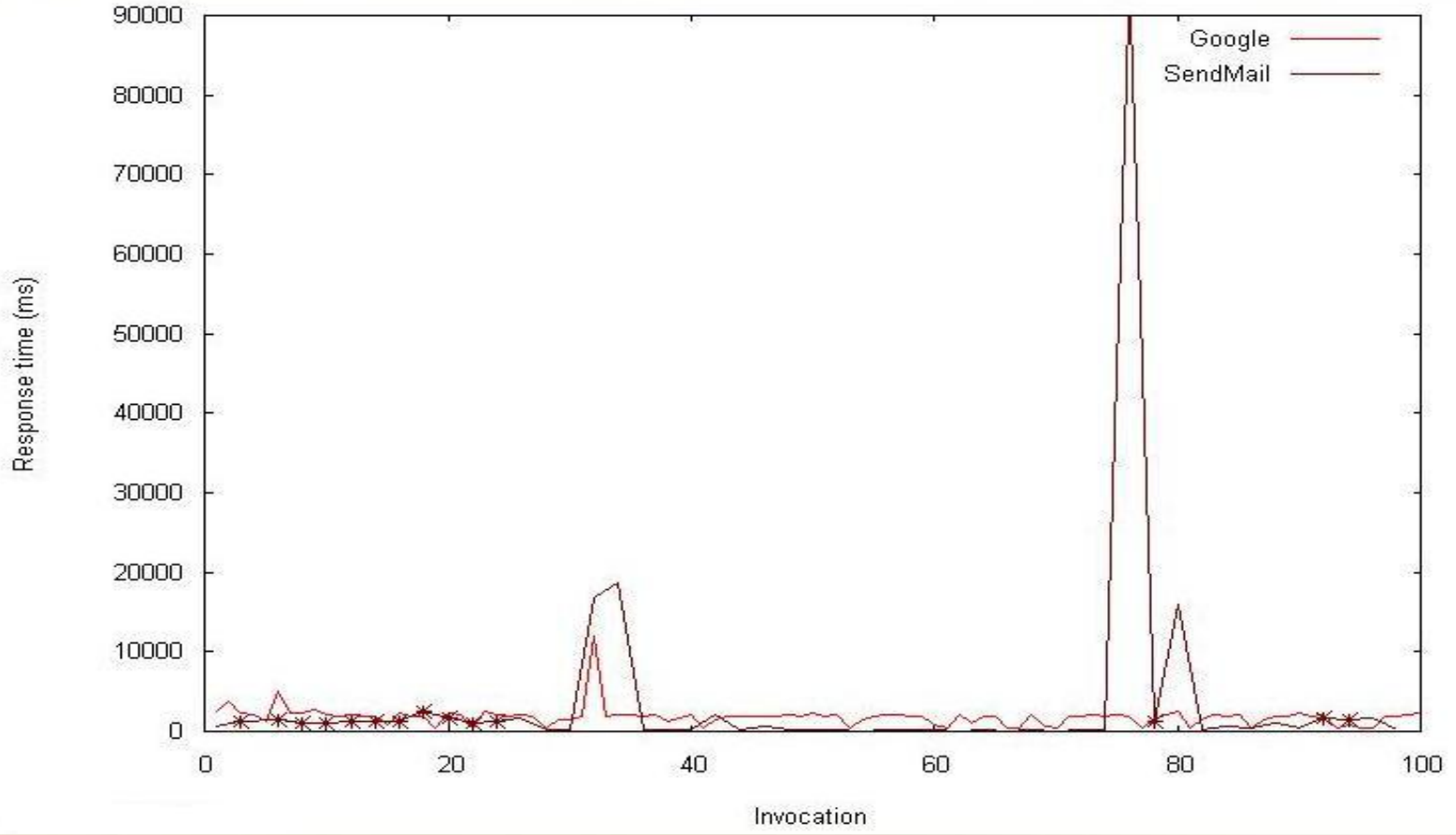
- **Qualities depend on**
  - **network layout**
  - **geographical location**
  - **time period, etc.**

- **Can we rely on QoS published by Provider?**
- **How Provider can evaluate QoS?**
- **Composite QoS metrics, different functions (min, max, average, etc.)**
- **How to meet Client's QoS-aware requests with provided Web Services?**

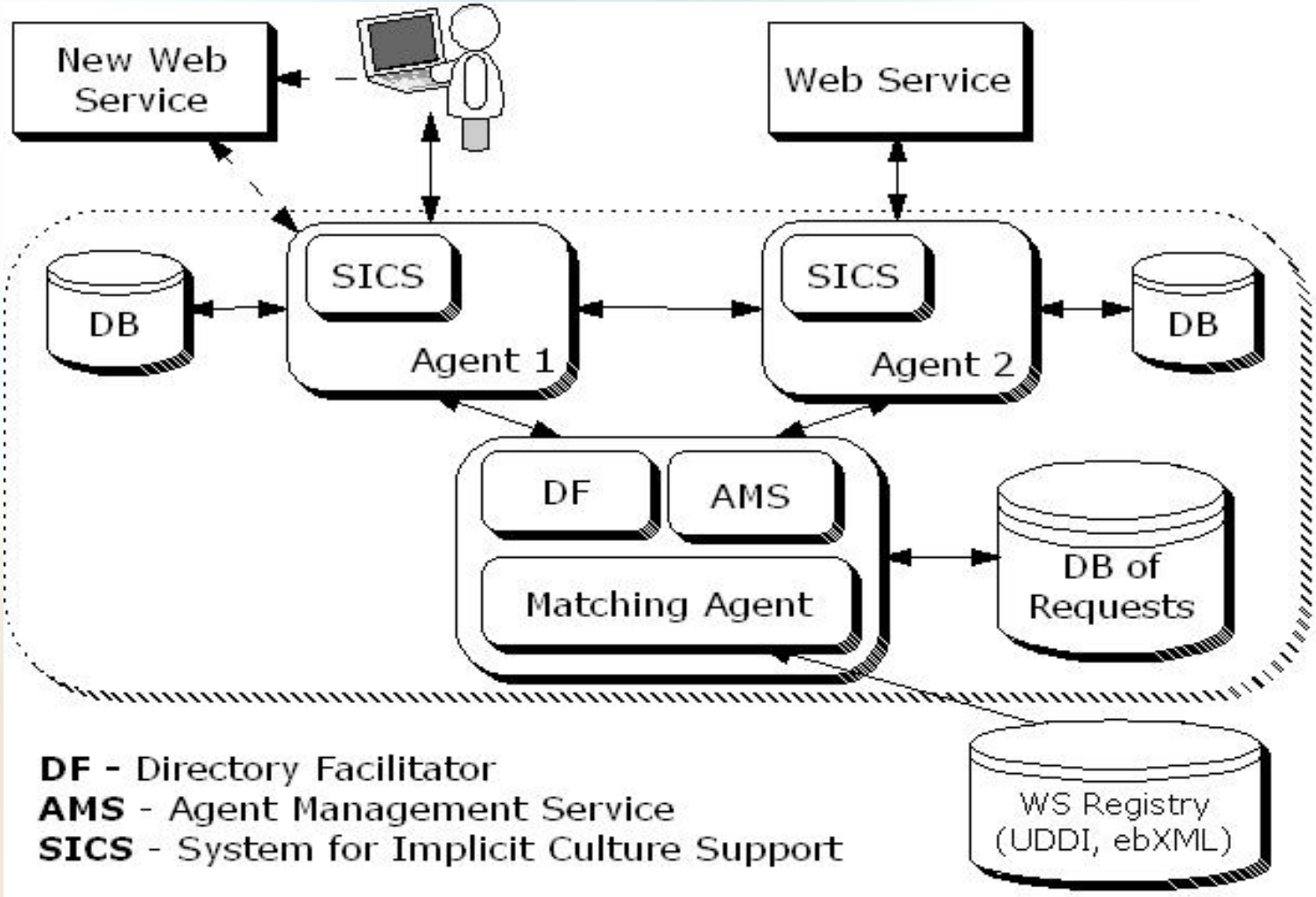
	Provider	Client		Agency
		Monitoring	Feedback	
Reliable	-	+	-	+
Objective	-	+	-	-
Recent	-	+	+	-
Free of charge	+	+	+	-
Low overheads	+	-	-	+
WS composition	-	+	-	-

QoS Issues

- **Example of QoS Monitoring: invocations of web services with the same input parameters in different time periods.**
  - **Response time**
  - **Availability**



Framework Implementation



Implicit Culture **- "to make the agent behave as a member of the group would do" [Birukov05]**

Framework Implementation

- **Queries**
  - **WSDL specifications**
  - **Structured XML-based queries**
    - **Documentation: *Search engine***
    - **Operation: *FindDocument***
      - **Input parameters:**
        - **Name: *query***
        - **Type: *String***
- **Implementation**
  - **Agent-based framework: JADE (Java Agent Development Framework)**
  - **WS Matching: *wsdl4j*, *Lucene* (for indexing)**
- **Evaluation: averge precision**

Future Work

- **Matching Algorithm Evaluation**
  - **WordNet**
  - **User vocabularies**
- **Framework Implementation**
- **Evaluation and Optimization**
- **Peer-to-Peer**
  - **Agents not only provide QoS but also match WSs**
- **Semantic Web services**
  - **WS Matching using ontologies**

Questions from reviewers

- **Compare UDDI with ebXML registry**
- **Are you going to consider a service specification as a document?**
- **Quality can vary (the registry must be updated periodically). Do you propose a solution to that?**
- **Description of the quality (WS-Policy)**
- **Look to the Semantic Web (OWL-S, WSMO)**

UDDI vs. ebXML

Requirement	UDDI (v3)	EbXML Registry (v3)
Service registration	+	+
Basic service discovery	+	+
Ability to store WSDL in repository	-	+
Manage life-cycle of WSDL	-	+
Access control for WSDL	-	+
Cataloging for WSDL	-	+
Discovery of WSDL	-	+
Pre-defined queries	+	+
Queries with predicate support	-	+
SQL queries	-	+
XML-based queries	-	+
Parameterized queries	-	+
Federated queries	-	+
Pre-defined taxonomies	+	+
User-defined taxonomies	-	+
Ontology support	-	+

References

[Birukov05] Birukov, A., Blanzieri, E., Giorgini, P.: ”Implicit: An agent-based recommendation system for web search”, Proceedings of the 4th International Conference on Autonomous Agents and Multi-Agent Systems, 2005, pp. 618–624.

[Dong04] Dong, X.L. et al.: ”Similarity search for web services”, Proceedings of VLDB, 2004.

[Maximilien02] Maximilien, E.M., Singh, M.P.: ”Conceptual Model of Web Service Reputation”, SIGMOD Record, Vol. 31, No. 4, Dec. 2002, pp. 36-41.

[Sajjanhar04] Sajjanhar, A., Hou, J., Zhang, Y.: ”Algorithm for Web Services Matching”, Proceedings APWeb 2004, LNCS 3007, Springer Verlag, 2004, pp. 665–670.

[Sajjanhar05] Stroulia, E., Wang, Y.: ”Structural and semantic matching for accessing web service similarity”, International Journal of Cooperative Information Systems, Vol. 14, No. 4, 2005, pp. 407–437.