University of Waterloo

Semantic-Based Context-Aware Service Discovery in Pervasive Computing Environments

Abdur-Rahman El-Sayed and James P. Black
David R. Cheriton School of Computer Science
University of Waterloo
Canada

IEEE International Workshop on *Services Integration in Pervasive Environments (SIPE)*, June 29, 2006, Lyon, France

In conjunction with IEEE ICPS'06

Outline

- Motivation
- Background
- Discovery Architecture
  - Overview
  - Service Description and Advertisement
  - Service Request and Matchmaking
  - Service Selection and Ranking
  - Service Invocation
- Implementation
- Conclusions

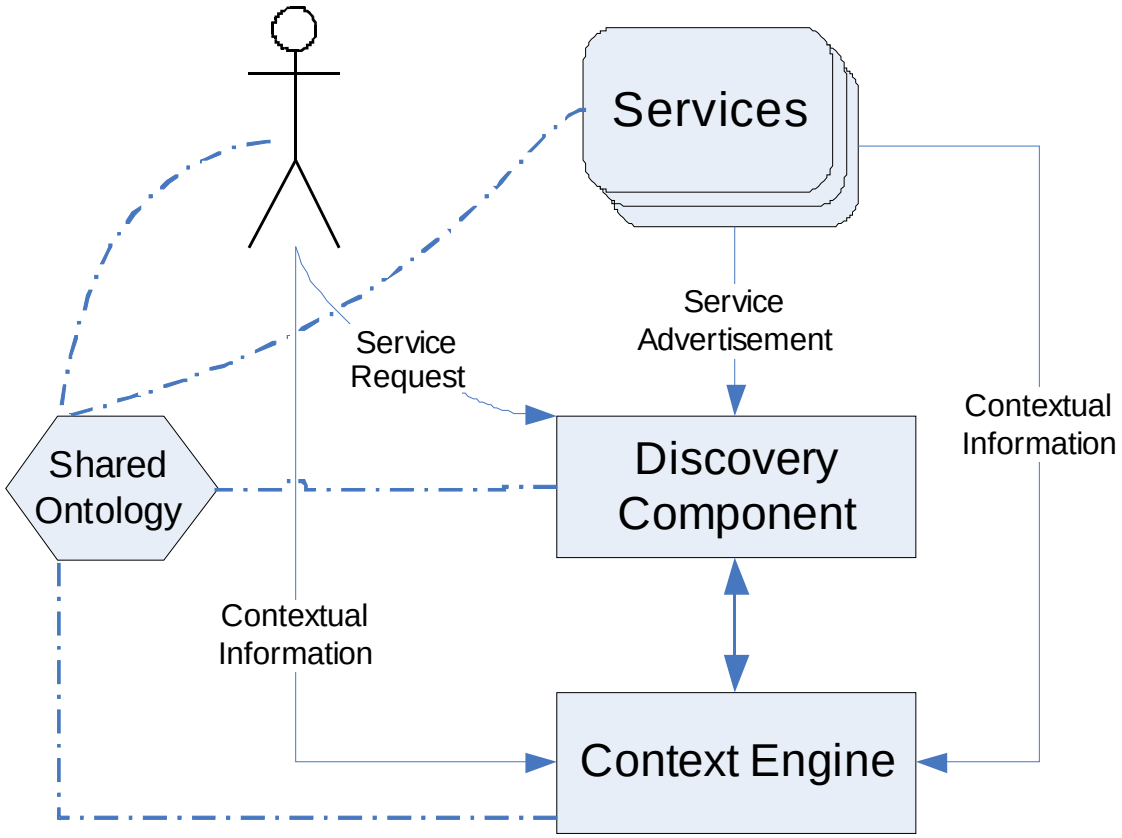
Motivation

- Current discovery protocols are not suitable for pervasive-computing environments
- (1) No use of contextual information
  - Fail to discover the most relevant/appropriate services
  - Nearest and least-loaded printer example
- (2) Reliance on a syntactic representation of services
  - Syntactically different but semantically equivalent
  - Syntactically equivalent but semantically different

Background – Impress Project

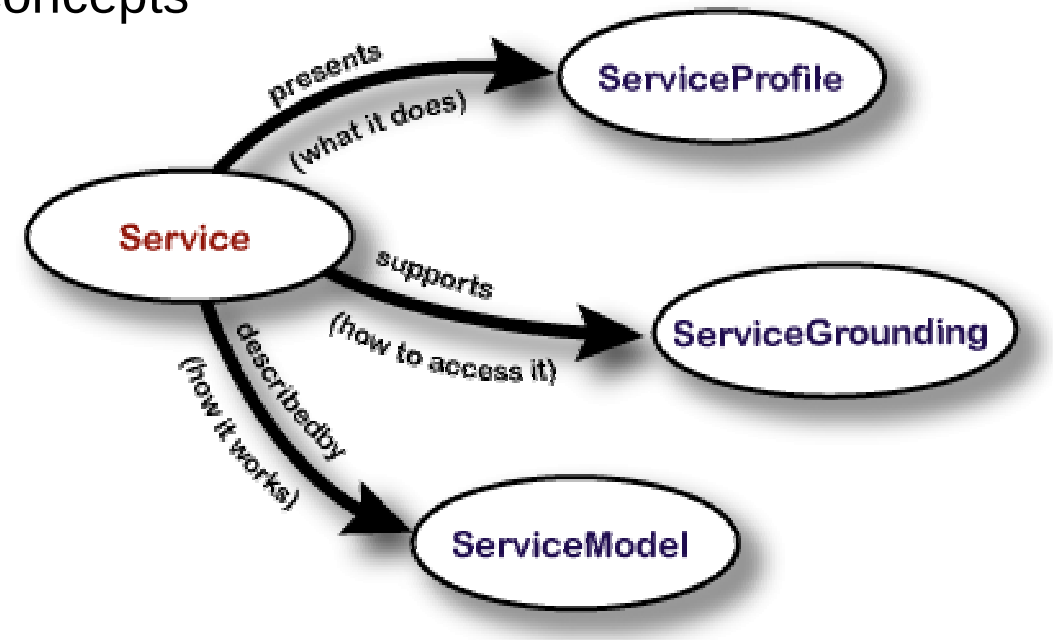
- Goal: turning the vision of ubiquitous computing into a reality
- Based on Jabber
  - XML-Based Instant-messaging protocol
  - Open-Source, Standards-based (XMPP), Extensible, and Secure
  - Proven to be deployable

Discovery Architecture – Overview



The Web Services Ontology (OWL-S)

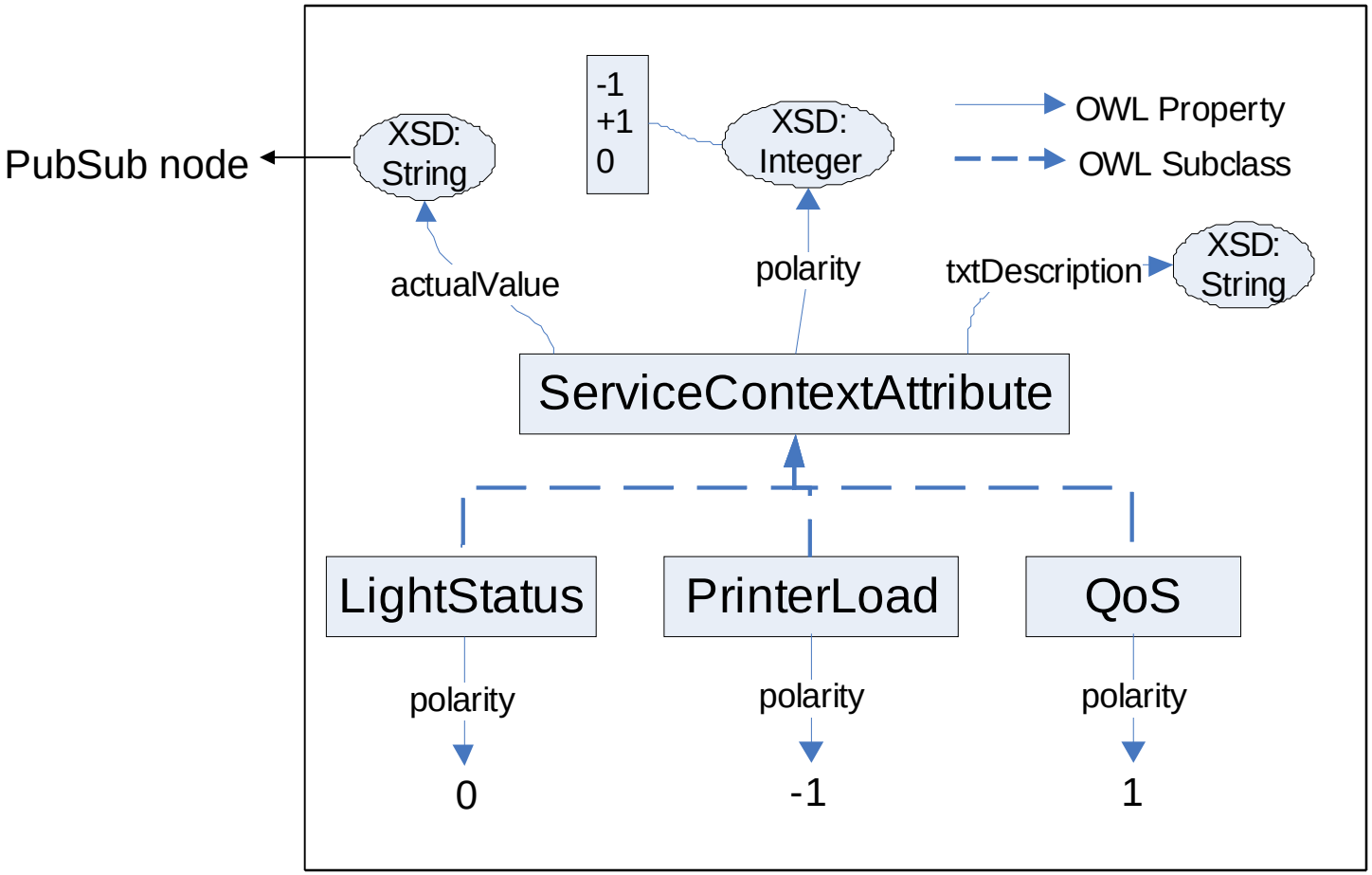
- Describes the properties and capabilities of Web Services
- Goal: autonomous service discovery, composition, and invocation
- Composed of three main concepts



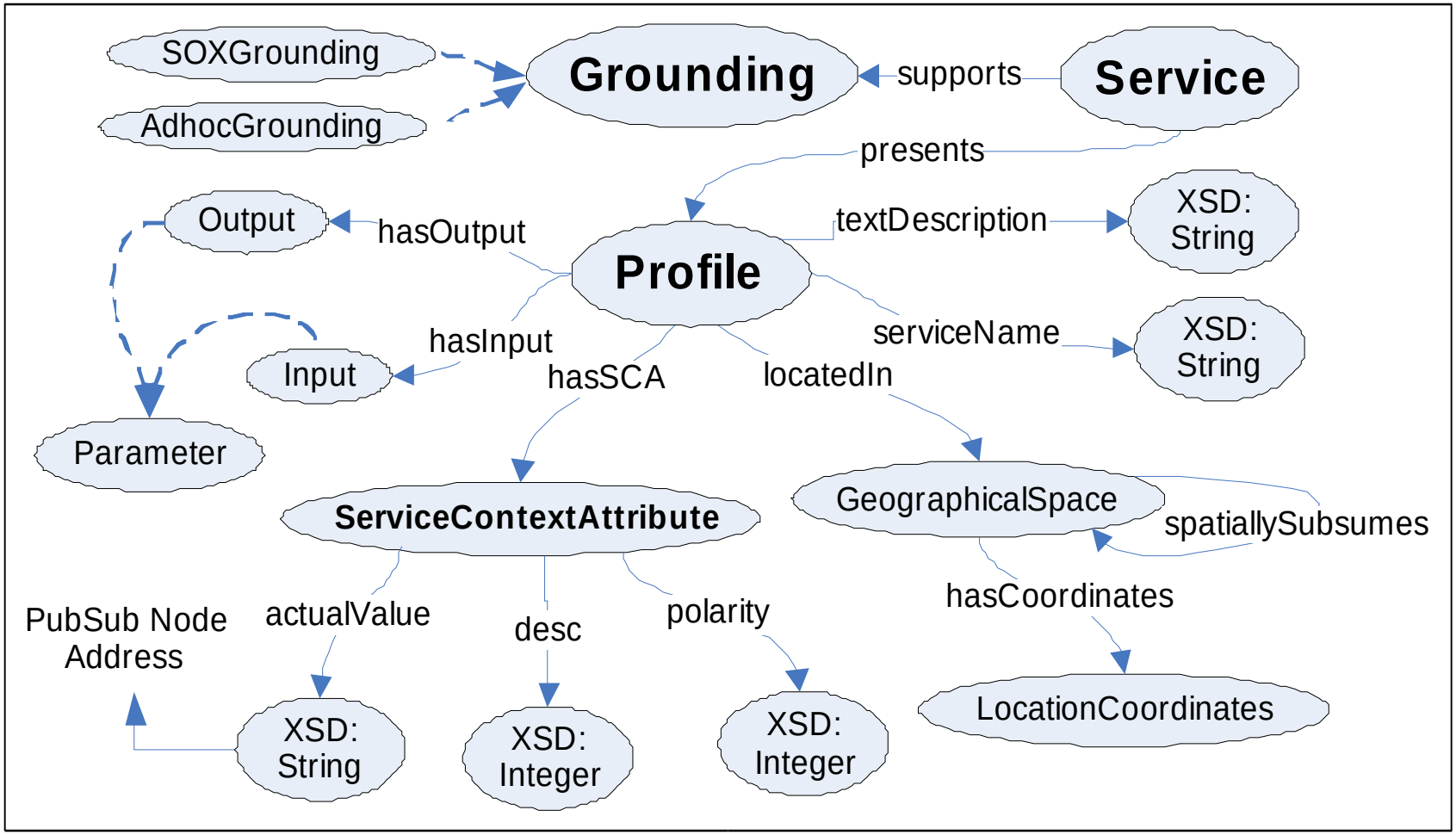
Service Description

- Problematic issues with OWL-S
  - Location limitation
  - Context-awareness
- Extend OWL-S with new concepts
  - ServiceProfile
  - ServiceGrounding
  - Dynamic nature of contextual information
    - Store actual values in a PubSub system (Context Engine)
- Combine with other ontologies
  - SOUPA
  - Re-use knowledge

ServiceContextAttribute Class

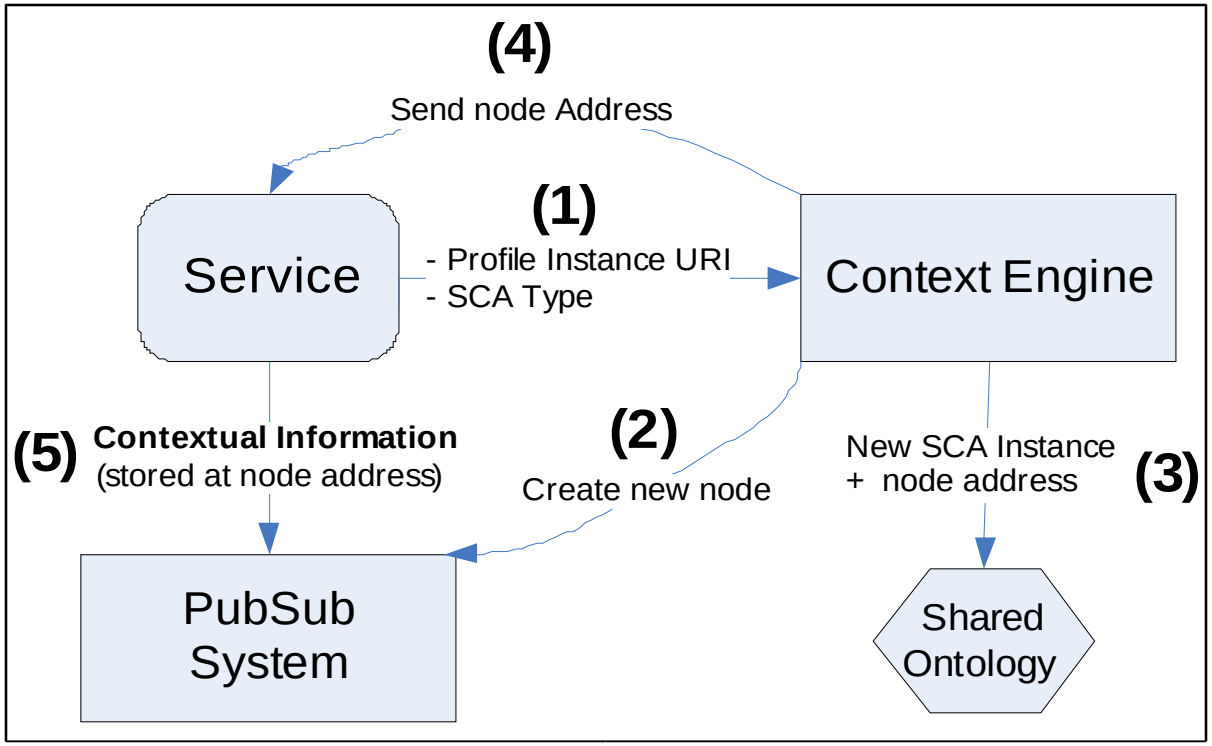


Exended ServiceProfile Class



Service Advertisement

- OWL construction is time-consuming
- GUI tool
- Context information of services
  - Push mode
  - Pull mode



Service Request

- Encoded in a Jabber/XMPP XML message
- Format of service request
  - Desired ServiceProfile
  - Can include the desired location
  - Can include the desired inputs/outputs

Service Request – Examples (1)

Service Request for a (Printer) Service located in the (University of Waterloo)

```
<iq type=get from=uwuser@jabber.org  to=discovery.jabber.org>

    <query xmlns=http://impress.uw.ca/#discovery-request>

        <profile>http://impress.uw.ca/#Printer</profile>

        <location>http://impress.uw.ca/#WaterlooUniv</location>

    </query>
</iq>
```

Service Request – Examples (2)

Service Request for a software service with specific inputs/outputs

```
<iq type='get' from=uwuser@jabber.org  to='discovery.jabber.org'>

 <query xmlns='http://impress.com/discover#'>

  <inputs>
    <input>http://impress.uw.ca/#Book</input>
 </inputs>

 <outputs>
    <output>http://impress.uw.ca/#Price</output>
  </outputs>

</query>
</iq>
```

Service Matchmaking

- *Step #1*: Parse request and expand it (reasoning)
  - Expand location and profile
- *Step #2*: Retrieve user's contextual information
  - User's location & preferences
- *Step #3*: Construct **SPARQL** query to find matching services
- *Step #4*: Rank matching services
  - Based on the dynamic contextual information of services

Sample SPARQL Query

- SPARQL query to find Printing services located in University of Waterloo

```
PREFIX impress: <http://impress.uw.ca/#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX spc:<http://pervasive.semanticweb.org/2004/06/space>

SELECT ?service
WHERE
{
  {   (?service, rdf:type, impress:Printer) .
      (?service, impress:locatedIn, impress:WaterlooUniv)
  }
UNION
  {  (?service, rdf:type, impress:Printer) .
     (?service, impress:locatedIn, ?x) .
     (?x, spc:spatiallySubsumedBy, impress:WaterlooUniv)
  }
}
```

Ranking Strategy

- ***Step #1***: Retrieve all contextual attributes associated with each matching service
  - ServiceProfile & Context Engine
- ***Step #2***: Construct a ranking table for each matching service
  - Currently, weights are equal

Ranking table for a matching printing service

Contextual Attribute	Value	Polarity	Weight
PrinterLoad	13	-1	0.50
Location (Distance)	21	-1	0.50

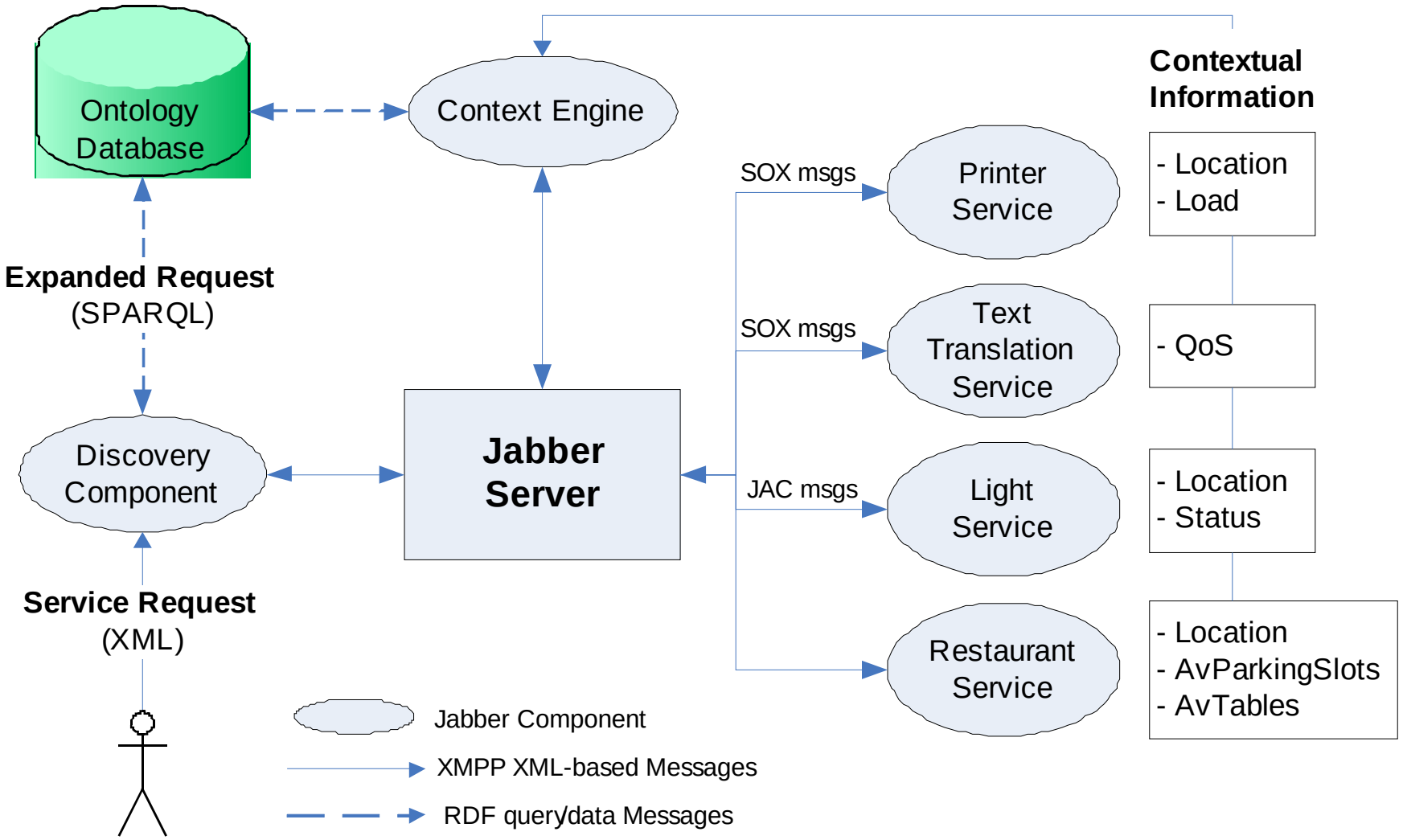
Ranking Strategy (cont')

- ***Step #3***: Compute a score for each matching service
  - n = total number of contextual attributes
  - V = actual value of the *i*th attribute
  - P = polarity of the *i*th attribute
  - W = weight of the *i*th attribute = 1 / n
  - Score = S = $\sum_{i=1}^{n} Vi \times Pi \times Wi$

Service Invocation

- Jabber/XMPP Invocation schemes
  - Jabber Adhoc commands (JAC)
    - Convenient for simple command-based services
  - Jabber RPC extension
  - Soap over XMPP (SOX)
    - Complex
    - Suitable for software-based services
    - WSDL interface
- AdhocGrounding and SOXGrounding
  - Provide a mapping from an abstract to a concrete specification of service capabilities

Prototype Overview



Contributions

- OWL-based ontology to facilitate context-aware discovery
- Discovery architecture
  - capability-based search
  - exploits contextual information
  - discovers and ranks most appropriate services
- Development of several services, different invocation schemes

Future Work

- Support wide-area service discovery
- Enable users to specify discovery preferences and weights in an unobtrusive manner
- Capture user requests using an appropriate HCI mechanism
- Scalability/Performance Tests

University of Waterloo

Semantic-Based Context-Aware Service Discovery in Pervasive Computing Environments

Abdur-Rahman El-Sayed and James P. Black
David R. Cheriton School of Computer Science
University of Waterloo
Canada

IEEE International Workshop on *Services Integration in Pervasive Environments (SIPE)*, June 29, 2006, Lyon, France

In conjunction with IEEE ICPS'06