

# Exporting DF Services

Agentcities.NET iD2  
Lisbon, 9-10th September, 2002

Margaret Lyell  
*MITRE Corporation*



# Outline

- Description
  - Background on the Directory Facilitator
  - Software Agents and DF Registration
  - Proposed DF Exports
- SOAP-based Protocol for Exporting DF
- Behavior
  - Agent Behavior
- Registration in Non-Agent Repositories
- Tasks To Do

# Part 1

## Description

# DF Background/1

- Software agents that are registered on a particular agent platform have the option of registering with the Directory Facilitator (DF)
- The DF provides a ‘yellow pages’ service
- With registration, a software agent can provide the following descriptive information on:
  - The conversation /interaction protocols that it supports (Set of String)
  - The content languages that it supports (Set of String)
  - The ontologies that it can utilize (Set of String)
  - The services it provides and descriptions of these services (Set of Service Description)
  - Its name (Agent Identifier)

# DF Background/2

- Moreover, the Service Description information can include:
  - The name of the service (String)
  - The type of the service (String)
  - A list of the interaction protocols that the service supports (Set of String)
  - A list of the content languages that the service supports (Set of String)
  - A list of the ontologies it can utilize (Set of String)
  - A list of the properties that describe the service (Set of Property ({name,value} pairs))
- The name and value parameters should be typed:
  - The owner of the service (String)

# DF Background/3

- The Agent Identifier (AIDs) includes:
  - The symbolic name of the agent name (Word)
  - A sequence of ordered transport address where the agent can be contacted, with the order implying a preference (Sequence of URL)
  - A list of name resolvers (Sequence of AIDs)
- NOTE: Only the first element is mandatory information in the AID

# DF Registration

- A software agent has the option of providing partial information to the DF in support of its registration:
  - For example, a software agent can elect to provide self-identifier information and service property descriptive information while failing to provide any information as to the ontologies, languages or interaction protocols that it supports
- A DF can be searched:
  - An agent can search the DF
  - The search can be made subject to search constraints
- The DF offers software agents:
  - The service of advertising in its yellow pages
  - The service of searching its listings

# Proposed DF Exports

- The service of the DF that is to be exported off of the agent platform is the service of searching its listings:
  - search



# Part 2

## Protocol for Exporting DF

# Protocol for Exporting DF/1

- Information held by the DF that can be exported will be exported via the SOAP protocol
- Access to the exported search service of the DF will be via a SOAP message
- The appropriate endpoint will be defined, in a manner befitting the type of SOAP binding used
- A SOAP Processor will be responsible for:
  - Taking the SOAP message
  - Building an ACL message
  - Sending the ACL message to the DF
  - Returning the result to the requestor

# Protocol for Exporting DF/2

- The SOAP message will include:
  - The method will be `searchDF`
  - The first parameter for this method is the complex type of `DFAgentDescription`:
    - it is composed of complex types
    - numerous elements of the complex type `DFAgentDescription` will be 'null' in the request. That is, only certain items may be specified in the search
    - the search should find all agents, along with their full agent descriptions, that match the search requirements
  - The second parameter gives the maximum number of agents/agent descriptions that the user wishes to have returned
- The method and the parameters are listed in the body of the SOAP envelope

# Protocol for Exporting DF/3

- The response also given in the body of the SOAP envelope will contain at least one item:
  - The first is of type 'int' and denotes the number of matching agents/descriptions that were returned
  - This number can be less than the number that were found by the search task
    - If the number is zero, then no agents matching the search parameters were found
    - If the number is greater than zero, then that number of items of complex type `DFAgentDescriptions` will be found in the body

# Protocol for Exporting DF/4

- Schema fragments for the relevant complex types are to be used as elements in the SOAP message:
  - The namespace is the FIPA namespace or `fipans`
- The Complex types include:
  - `PropertyTemplate`
  - `ServiceDescription`
  - `DFAgentDescription`

# Property Template

```
<complexType name="PropertyTemplate"
    type="fipans:PropertyTemplate"/>
<!-- From FIPA specifications-->
<sequence>
    <element name="property"
        minOccurs="1"
        maxOccurs="unbounded" />
        <complexType>
            <sequence>
                <element name="name" type="xsd:string"/>
                <element name="value" type="xsd:string"/>
            </sequence>
        </complexType>
    </element>
</sequence>
</complexType>
```

# Service Description

```
<element name="ServiceDescription"
  type="fipans:ServiceDescription" />
<complexType name="ServiceDescription">
  <!-- Elements of DF Service Description -->
  <element name="serviceName" type="xsd:string" />
  <element name="serviceType" type="xsd:string" />
  <element name="interactionProtocol"
    type="xsd:string" />
  <element name="ontology" type="xsd:string" />
  <element name="contentLanguage"
    type="xsd:string" />
  <element name="property"
    type="fipans:PropertyTemplate" />
  <element name="owner" type="xsd:string" />
</complexType>
</element>
```

# DF Agent Description

```
<element name="DFAgentDescription"
    type="fipans:DFAgentDescription" />
<complexType name="DFAgentDescription">
  <!-- Elements of DF Agent Description -->
  <element name="agentName" type="xsd:string" />
  <element name="interactionProtocol"
    type="xsd:string" />
  <element name="ontology" type="xsd:string" />
  <element name="contentLanguage"
    type="xsd:string" />
  <element name="serviceDescription"
    type="fipans:ServiceDescription" />
</complexType>
</element>
```



# Part 3 Behaviour

# Behaviour

- The nature of the SOAP binding is an implementation detail and will not be addressed here:
  - Note that SOAP bindings are not confined to HTTP transport, for example, a JMS binding is possible
- The DF receives the request for a search in the same manner as if the requestor were an agent on the agent platform
  - The DF responds with the search results in the usual manner
- With regard to implementation, the SOAP processor may have a software agent proxy (or be implemented as an agent)

# Agent Behavior

- An agent that wishes to offer a SOAP service should provide:
  - The information necessary to describe the service
  - The list of interaction protocols that the agent supports in offering the SOAP service
  - The list of ontologies that the agent supports in offering the SOAP service
  - Others?
- If the agent does not provide pertinent information as part of its `ServiceDescription`, then a potential client cannot assess the relevancy of the service being offered

# **Part 4**

## **Registration in Non-Agent Repositories**

# Non-Agent Repositories/1

- This is motivated by a scenario in which a potential (non-agent) client does not know of the existence of a DF to search, but is aware of a local UDDI registry
- Within the Web Services framework consisting of SOAP, WSDL and UDDI, the UDDI Registry component serves as a repository for services
- An XML schema is used to define the core information model in UDDI:
  1. Business
  2. Service
  3. Binding
  4. Information on specifications for services

# Non-Agent Repositories/2

- Defined data structures provide a uniform structure for this information, for example, services are described in the `BusinessService` entity:
  - This data structure contains a number of elements, including that of (any number of) `categoryBag`
  - It is through the use of `categoryBag` entries that descriptive information can be included

- Example:

```
<categoryBag>
  <keyedReference
    tModelKey=
      "uddi:ubr.uddi.org:categorization:geo3155-2"
    keyName="GEO:France"
    keyValue="FR" />
</categoryBag>
```

# Non-Agent Repositories/3

- Software agents that register services with the DF might want to structure the description of their services according to the information model for the UDDI

# Part 5

## Tasks To Do



# Tasks To Do/1

- An example of how a service description should be constructed to facilitate the mapping of a service description pertaining to a service offered by a software agent into the UDDI registry format should be constructed
- A discussion as to whether a mapping of a service description pertaining to a service offered by a software agent into the UDDI registry format should be standardized should occur

# Tasks To Do/2

- A discussion as to whether certain types of information pertaining to a Service Description should be standardized should occur. For example, should Quality of Service (QoS) information be included in the properties of a Service Description? If so, can a description of QoS be provided?