1

FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

2

3

4

5
FIPA Agent Discovery Service Specification

6

7

Document title	FIPA Agent Discovery Service Specification		
Document number	Version 1.2e	**Document source**	FIPA TC Ad Hoc
Document status	Preliminary	**Date of this status**	2003/10/20
Supersedes	None		
Contact	adhoc@fipa.org		

8

9

10

11

12

13

14

15

16

Foreword

The Foundation for Intelligent Physical Agents (FIPA) is an international organization that is dedicated to promoting the industry of intelligent agents by openly developing specifications supporting interoperability among agents and agent-based applications. This occurs through open collaboration among its member organizations, which are companies and universities that are active in the field of agents. FIPA makes the results of its activities available to all interested parties and intends to contribute its results to the appropriate formal standards bodies where appropriate.

The members of FIPA are individually and collectively committed to open competition in the development of agent-based applications, services and equipment. Membership in FIPA is open to any corporation and individual firm, partnership, governmental body or international organization without restriction. In particular, members are not bound to implement or use specific agent-based standards, recommendations and FIPA specifications by virtue of their participation in FIPA.

The FIPA specifications are developed through direct involvement of the FIPA membership. The status of a specification can be either Preliminary, Experimental, Standard, Deprecated or Obsolete. More detail about the process of specification may be found in the FIPA Document Policy [f-out-00000] and the FIPA Specifications Policy [f-out-00003]. A complete overview of the FIPA specifications and their current status may be found on the FIPA Web site.

FIPA is a non-profit association registered in Geneva, Switzerland. As of June 2003, the 50 members of FIPA represented many countries worldwide. Further information about FIPA as an organization, membership information, FIPA specifications and upcoming meetings may be found on the FIPA Web site at http://www.fipa.org/.

Contents

1   Scope

This document deals with the discovery of agents and their offered services in ad hoc[1] networks. The Agent Discovery Service (ADS) is specified in addition to the well known FIPA agent directory services Agent Management System and Directory Facilitator [FIPA00023]. This document contains specifications for:

- A reference model for the discovery process, based on different technologies, in ad hoc networks.

- The functionality and interface of the ADS.

---

[1] Ad hoc networks comprise both mobile and fixed networks.

2   Reference Model – Discovery in Ad Hoc Networks

79

80 "Discovery in ad hoc networks" refers to the discovery of agents hosted on agent platforms (AP) on remote
81 devices. Directory Facilitator (DF) federations according to [FIPA00023] enable an agent to discover agents on
82 remote devices without having to know any remote DF. DF federations for discovery in ad hoc networks only work
83 if network nodes do not repeatedly join or leave.

84

85 The Agent Discovery Service (ADS) provides discovery functionality in ad hoc networks, in which network nodes
86 join or leave more frequently or less frequently. It provides a high-level DF-like interface for agents, while taking
87 advantage of various discovery middleware (DM), depending on the underlying ad hoc technology. Figure 1
88 outlines how the ADS fits in the FIPA architecture.

89
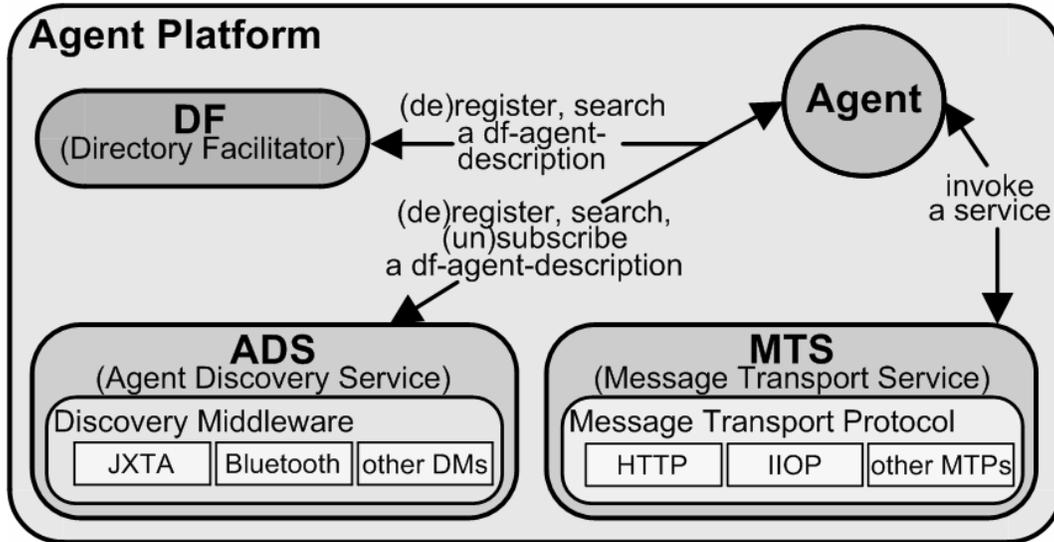
90   Figure 1: Reference Model of the Discovery Process in Ad Hoc Networks

91 An AP optionally hosts a DF. If an ADS is present on the AP, the DF should only be used for handling `df-agent-`
92 `descriptions` related to the local AP. The ADS should only be used by agents of the local AP for provision of
93 their `df-agent-description`s to the ad hoc network as well as for discovery of agents on remote devices in
94 the ad hoc network, i.e. the ADS allows local agents to be discovered by agents on remote devices and vice versa.
95 This means that the DF should provide a yellow pages service restricted to the scope of the local AP, and the ADS
96 provides a yellow pages service restricted to the scope of the ad hoc network. The ad hoc network accessible via
97 the ADS is a compound of all ad hoc networks supported by the maintained DM technologies, for instance JXTA or
98 Bluetooth.

99

100 Agent-to-agent communication is performed by using the Message Transport Service (MTS) according to
101 [FIPA00067]. The MTS might be extended with message transport protocols that are suited for the different ad hoc
102 networks, e.g. a JXTA transport or a Bluetooth OBEX transport.

3   Agent Discovery Service

The ADS is an optional entity, which is a reification of the Agent Directory Service in [FIPA00001] for discovery in ad hoc networks. The functionality it provides to agents is similar, but extended, to the one of a DF. The ADS maintains one or more DM technologies, each of which provides access to a certain ad hoc network.

3.1   ADS Functionality

The ADS supports functions that are similar to the `register`, `deregister` and `search` functions as specified in [FIPA00023] as part of the Agent Management Ontology function descriptions. Additionally, the ADS defines a subscribe/unsubscribe functionality. All of these functions make use of `df-agent-description`s specified in [FIPA00023] as part of the Agent Management Ontology object descriptions.

3.1.1   DM IDs

When using ADS functionality, agents may apply it either to all available DMs or to a subset. Hence, agents must be able to retrieve a list of all available DM IDs from the ADS. A DM ID is a string reserved for a single technology. Table 1 summarizes the currently available IDs.

DM ID String	Identified Technology	Notes
JXTA	JXTA	see [JXTA]
BT	Bluetooth	see [BT]

Table 1: DM IDs

3.1.2   Register and Deregister

The ADS supports the registration and the deregistration of `df-agent-description`s. If an agent registers with an ADS, it becomes discoverable for agents on remote devices. If an agent deregisters from an ADS, it is no longer discoverable for agents on remote devices. Each of these functionalities can be applied either to all available DMs or to a subset.

On registration with the ADS, an agent can specify a lease time, which is how long it would like the registration to be kept. When the lease time expires, the registration will be silently removed from the ADS.

The lease-time parameter of the `df-agent-description` is used by a DM to determine the lifetime of the `df-agent-description`. If no lease-time parameter exists in the `df-agent-description`, the lifetime is assumed to be unlimited. In this case, it is important to deregister the `df-agent-description` later on, in order to save system resources.

If the `df-agent-description` has already been registered, its lease-time is renewed according to the value of the `lease-time` parameter.

3.1.3   Search

The ADS supports a search that takes a `df-agent-description` search template and a specified duration. It returns within the specified duration all `df-agent-description`s found that match the `df-agent-description` search template. This functionality can be applied either to all available DMs or to a subset.

The search functionality may optionally constrain the number of returned results per agent platform. This is a means to limit both the network load as well as the processing load of a device. This functionality is similar to the one provided by the `max-results` parameter of the search-constraints frame in the Agent Management Ontology in [FIPA00023].

148   The matching criterion to determine the set of objects that satisfy the search criteria is exactly the same as
149   specified for the `search` function in [FIPA00023].

150   ### 3.1.4   Subscribe and Unsubscribe

151   The ADS provides a functionality to subscribe and unsubscribe `df-agent-description`s which match a `df-`
152   `agent-description` search template. From the time of subscription on, each newly registered `df-agent-`
153   `description` that satisfies the search criteria will be returned to the agent until unsubscription.[2] Each of these
154   functionalities can be applied either to all available DMs or to a subset.
155
156   The matching criterion, in order to determine the set of objects that satisfy the search criteria, is exactly the same
157   as specified for the `search` function in [FIPA00023].

---

[2] Note that, if the ADS should be implemented as an agent, it may be necessary to specify a FIPA Subscribe interaction protocol.

158 # 4   Discovery Middleware

159 Various technologies exist or will appear, which provide for discovery in ad hoc networks. Examples of such DM
160 technologies are JXTA and Bluetooth.
161
162 Each DM maintained by the ADS must be described in a FIPA specification of its own. In detail, such a DM
163 specification must describe how the DM functionality, which is imposed by the ADS functionality provided to
164 agents, can be realized on the basis of the respective technology.
165
166 This specification does not specify how the actual interface between the ADS and its DMs must look like.
167 However, functionality details of the DM, which are important to consider during DM specification, are summarized
168 next.
169

170 ## 4.1   Functionality of a Discovery Middleware

171 In order to support the ADS in offering agents the expected functionality, each DM in turn must provide similar
172 functionality to the ADS.
173

174 ### 4.1.1   DM ID

175 The ADS must be able to retrieve a DM's ID string (see also section 3.1.1). This enables the ADS to differentiate
176 between several DMs and to provide this information to the agents.
177

178 ### 4.1.2   Startup and Shutdown

179 The ADS must be able to start up and shut down a DM at runtime.
180

181 ### 4.1.3   Register and Deregister

182 The ADS must be able to register `df-agent-description`s within a DM and to deregister `df-agent-`
183 `description`s from the DM. See section 3.1.2 for the details of registration and deregistration.
184

185 ### 4.1.4   Search

186 The ADS must be able to search for `df-agent-description`s within a DM on the basis of a `df-agent-`
187 `description` search template. See section 3.1.3 for the details of the search.
188

189 ### 4.1.5   Subscribe and Unsubscribe

190 The ADS must be able to subscribe and unsubscribe `df-agent-description`s within a DM which match a `df-`
191 `agent-description` search template. See section 3.1.4 for the details of subscription and unsubscription.

192   **5 Agent Discovery Ontology**

193   **5.1 Object Descriptions**

194 The ADS makes use of the `df-agent-description` frame which is specified by [FIPA00023] as part of the
195 Agent Management Ontology object descriptions. Additionally the ADS requires other frames, which are part of the
196 Agent Discovery Ontology. This section describes the set of frames that represent the classes of objects in the
197 domain of discourse within the framework of the `fipa-agent-discovery` ontology.
198
199 This ontology does not specify any specific positional order to encode the parameters of the objects. Therefore, it
200 is required to encode objects in SL by specifying both the parameter name and the parameter value (see Section
201 3.6 of [FIPA00008]).
202
203 The following terms are used to describe the objects of the domain:
204

205 - **Frame**. This is the mandatory name of this entity that must be used to represent each instance of this
206   class.
207

208 - **Ontology**. This is the name of the ontology, whose domain of discourse includes the parameters
209   described in the table.
210

211 - **Parameter**. This is the mandatory name of a parameter of this frame.
212

213 - **Description**. This is a natural language description of the semantics of each parameter.
214

215 - **Presence**. This indicates whether each parameter is mandatory or optional.
216

217 - **Type**. This is the type of the values of the parameter: Integer, Word, String, URL, Term, Set or Sequence.
218

219 - **Reserved Values**. This is a list of FIPA-defined constants that can assume values for this parameter.
220

221   **5.1.1 DM Constraints**

222 This type of object represents a set of DMs, on which the different functions of the ADS should be applied.
223

Frame	`dm-constraints`			
Ontology	`fipa-agent-discovery`			
Parameter	**Description**	**Presence**	**Type**	**Reserved Values**
`dms`	A list of DMs, which should be used to invoke an ADS function.	Optional	Set of `string`	JXTA BT

224

225   **5.1.2 Search Constraints**

226 This type of object represents a set of constraints to limit the function of searching within the ADS.
227

Frame	`search-constraints`			
Ontology	`fipa-agent-discovery`			
Parameter	**Description**	**Presence**	**Type**	**Reserved Values**
`timeout`	The time in milliseconds to wait for answers, after a search query has been emitted.	Mandatory	`integer`	
`max-results`	The maximum number of returned results per agent	Optional	`integer`	

	platform.			

| dms | A list of DMs, which should be used for the search. | Optional | dm-constraints | |

228

5.2   Function Descriptions

230 The following tables define usage and semantics of the functions that are part of the fipa-agent-discovery
231 ontology and that are supported by the ADS.
232
233 This ontology does not specify any specific positional order to encode the parameters of the objects. Therefore, it
234 is required to encode objects in SL by specifying both the parameter name and the parameter value (see section
235 3.6 of [FIPA00008]).
236
237 The following terms are used to describe the functions of the fipa-agent-discovery domain:
238
239   • **Function**. This is the symbol that identifies the function in the ontology.
240
241   • **Ontology**. This is the name of the ontology, whose domain of discourse includes the function described in
242   the table.
243
244   • **Supported by**. This is the type of agent that supports this function.
245
246   • **Description**. This is a natural language description of the semantics of the function.
247
248   • **Domain**. This indicates the domain over which the function is defined. The arguments passed to the
249   function must belong to the set identified by the domain.
250
251   • **Range**. This indicates the range to which the function maps the symbols of the domain. The result of the
252   function is a symbol belonging to the set identified by the range.
253
254   • **Arity**. This indicates the number of arguments that a function takes. If a function can take an arbitrary
255   number of arguments, then its arity is undefined.
256

5.2.1   Registration of a df-agent-description with the ADS

Function	register
Ontology	fipa-agent-discovery
Supported by	ADS
Description	The execution of this function has the effect of registering a new df-agent-description into the knowledge base of the ADS. The df-agent-description supplied must include a valid AID. To prevent the registration of the df-agent-description with all available DMs, the DM-IDs of the desired DMs can be passed.
Domain	df-agent-description $\times^3$ dm-constraints
Range	The execution of this function results in a change of the state, but it has no explicit result. Therefore there is no range set.
Arity	2

258

5.2.2   Deregistration of a df-agent-description from the ADS

Function	deregister
Ontology	fipa-agent-discovery

---

[3] Where $\times$ is the Cartesian product.

Supported by	ADS
Description	An agent may deregister a `df-agent-description` in order to remove all of its parameters from the ADS. The `df-agent-description` supplied must include a valid AID. To prevent the deregistration of the `df-agent-description` from all available DMs, the DM-IDs of the desired DMs can be passed.
Domain	`df-agent-description` × `dm-constraints`
Range	The execution of this function results in a change of the state, but it has no explicit result. Therefore there is no range set.
Arity	2

260

261   **5.2.3   Search for `df-agent-description` Registrations within the ADS**

Function	`search`
Ontology	`fipa-agent-discovery`
Supported by	ADS
Description	An agent may search for certain `df-agent-description`s by passing a `df-agent-description` template to the ADS. A successful search can return one or more `df-agent-description`s that satisfy the search criteria and returned within a fixed amount of time. A null set is returned when no `df-agent-description` entries satisfy the search criteria. A null set is also returned when the defined search duration is exceeded, even if some results would have been received later on. To prevent a search on all available DMs, the DM-IDs of the desired DMs can be passed. Further, the maximum number of returned results per agent platform can be defined.
Domain	`df-agent-description` × `search-constraints`
Range	Set of `df-agent-description`s
Arity	2

262

263   **5.2.4   Subscribe for `df-agent-description` Registrations within the ADS**

Function	`subscribe`
Ontology	`fipa-agent-discovery`
Supported by	ADS
Description	An agent may subscribe to receive registered `df-agent-description`s by passing a `df-agent-description` template to the ADS. From the time of subscription on, each newly registered `df-agent-description` that satisfies the search criteria will be returned to the agent until unsubscription.[4] To prevent a subscription on all available DMs, the DM-IDs of the desired DMs can be passed.
Domain	`df-agent-description` × `dm-constraints`
Range	The execution of this function results in a change of the state, but it has no explicit result. Therefore there is no range set. But, due to the asynchronous nature of the subscription, a set of matching `df-agent-description`s may be returned to the subscribing agent at later points in time.
Arity	2

264

265   **5.2.5   Unsubscribe from `df-agent-description` Registrations within the ADS**

Function	`unsubscribe`
Ontology	`fipa-agent-discovery`
Supported by	ADS
Description	An agent may cancel its subscription to receive registered `df-agent-`

---

[4] Note that, if the ADS should be implemented as an agent, it may be necessary to specify a FIPA Subscribe interaction protocol.

	descriptions by passing to the ADS the df-agent-description template, which was used for subscription. To prevent an unsubscription on all available DMs, the DM-IDs of the desired DMs can be passed.
Domain	df-agent-description × dm-constraints
Range	The execution of this function results in a change of the state, but it has no explicit result. Therefore there is no range set.
Arity	2

266

267 **5.2.6   Retrieve the IDs of DMs hosted by the ADS**

Function	get-dms
Ontology	fipa-agent-discovery
Supported by	ADS
Description	An agent may retrieve the list of IDs of all available DMs offered by the ADS.
Domain	None
Range	Set of DM IDs
Arity	0

268

269 **5.3   Exceptions**

270 Under some circumstances, an exception can be generated, for example, when an AID that has been already
271 registered is re-registered. This specification makes use of those exceptions as described in [FIPA00023] in the
272 Agent Management Ontology.

273

6   References

274

275 [FIPA00001]   FIPA Abstract Architecture Specification. Foundation for Intelligent Physical Agents, 2002.
276   `http://www.fipa.org/specs/fipa00001/`
277 [FIPA00008]   FIPA SL Content Language Specification. Foundation for Intelligent Physical Agents, 2002.
278   `http://www.fipa.org/specs/fipa00008/`
279 [FIPA00023]   FIPA Agent Management Specification. Foundation for Intelligent Physical Agents, 2002.
280   `http://www.fipa.org/specs/fipa00023/`
281 [FIPA00067]   FIPA Agent Message Transport Service Specification. Foundation for Intelligent Physical Agents,
282   2002.
283   `http://www.fipa.org/specs/fipa00067/`
284 [JXTA]   Project JXTA.
285   `http://www.jxta.org`
286 [BT]   Bluetooth.
287   `http://www.bluetooth.org/`

288 ## 7   Informative Annex A – Control Flow Example

289 To further clarify how ADS and DMs work together, an example will be given below. Figure 2 outlines the control
290 flow of an ADS based discovery process.
291
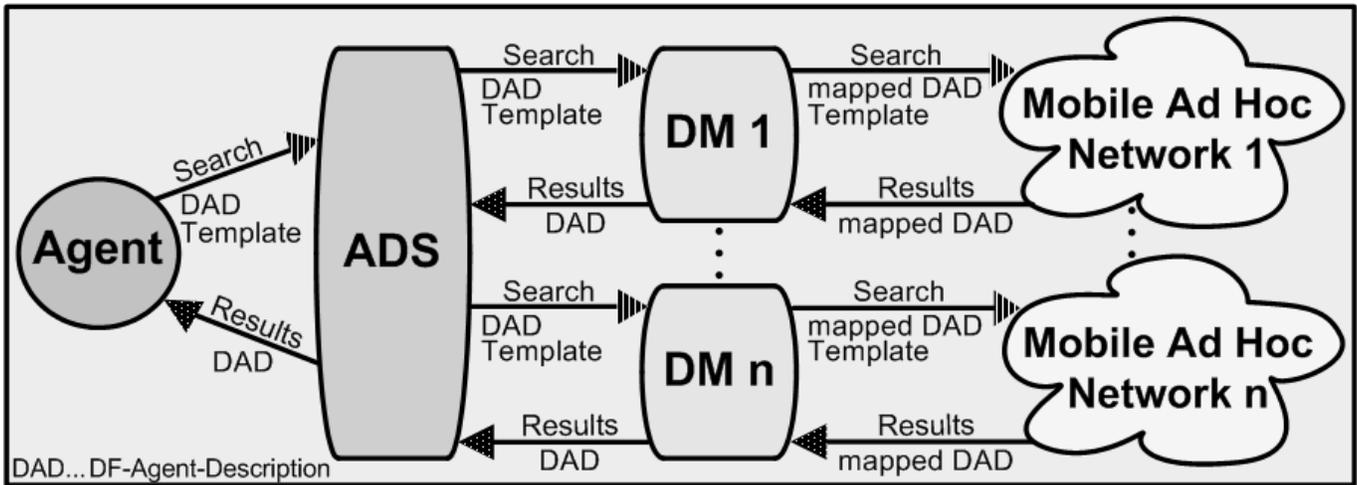
292   Figure 2: Control Flow of an ADS based Discovery

293 For searching a certain agent service, first the agent must compose a `df-agent-description` search template
294 describing the service to search. (To find a local service, the agent may ask an existing DF with the same `df-`
295 `agent-description`.)
296

297 To find the service in one or more ad hoc networks, the agent uses the search functionality of the ADS by
298 providing the composed `df-agent-description` search template and the time within which the search must be
299 finished. Eventually the agent might also specify special DM(s) which should be used for this search. This can be
300 done after the agent has queried the ADS for all currently available DMs. Otherwise, the ADS will use all available
301 DMs by default.
302

303 The ADS now uses the search functionality of each available DM, in turn by providing the `df-agent-`
304 `description` search template.
305

306 Each invoked DM maps the `df-agent-description` search template to the appropriate representation of the
307 corresponding technology and performs the search in the corresponding ad hoc network.
308

309 All available `df-agent-description`s, which have previously been registered with the ADS, are represented in
310 the DM's appropriate form and are discoverable by agents on remote devices.
311

312 The results of such a search are matching `df-agent-description`s in the appropriate representation of the
313 DM. The DM is now mapping the results back to real `df-agent-description`s and returns them to the ADS.
314

315 The ADS is collecting the resulting `df-agent-description`s of all invoked DMs and is returning them to the
316 calling agent.