

# FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

## FIPA Nomadic Application Support Specification

<b>Document title</b>	FIPA Nomadic Application Support Specification		
<b>Document number</b>	SI00014H	<b>Document source</b>	FIPA TC Nomadic Application Support
<b>Document status</b>	Standard	<b>Date of this status</b>	2002/12/03
<b>Supersedes</b>	FIPA00062, FIPA00063, FIPA00065, FIPA00066		
<b>Contact</b>	fab@fipa.org		
<b>Change history</b>	See <i>Informative Annex A — ChangeLog</i>		

© 1996-2002 Foundation for Intelligent Physical Agents  
<http://www.fipa.org/>  
Geneva, Switzerland

### Notice

Use of the technologies described in this specification may infringe patents, copyrights or other intellectual property rights of FIPA Members and non-members. Nothing in this specification should be construed as granting permission to use any of the technologies described. Anyone planning to make use of technology covered by the intellectual property rights of others should first obtain permission from the holder(s) of the rights. FIPA strongly encourages anyone implementing any part of this specification to determine first whether part(s) sought to be implemented are covered by the intellectual property of others, and, if so, to obtain appropriate licenses or other permission from the holder(s) of such intellectual property prior to implementation. This specification is subject to change without notice. Neither FIPA nor any of its Members accept any responsibility whatsoever for damages or liability, direct or consequential, which may result from the use of this specification.

## 20 **Foreword**

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

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

31 The FIPA specifications are developed through direct involvement of the FIPA membership. The status of a  
32 specification can be either Preliminary, Experimental, Standard, Deprecated or Obsolete. More detail about the process  
33 of specification may be found in the FIPA Document Policy [f-out-00000] and the FIPA Specifications Policy [f-out-  
34 00003]. A complete overview of the FIPA specifications and their current status may be found on the FIPA Web site.

35 FIPA is a non-profit association registered in Geneva, Switzerland. As of June 2002, the 56 members of FIPA  
36 represented many countries worldwide. Further information about FIPA as an organization, membership information,  
37 FIPA specifications and upcoming meetings may be found on the FIPA Web site at <http://www.fipa.org/>.

## 38 Contents

39	1	Scope.....	1
40	2	General Analysis.....	2
41	2.1	Overview .....	2
42	2.2	Monitoring and Controlling Quality of Service .....	2
43	2.3	Negotiation of Message Transport Requirements.....	4
44	2.3.1	Negotiation about Message Transport Protocols .....	4
45	2.3.2	Negotiation about Message Representation .....	4
46	3	Nomadic Application Support Ontology.....	5
47	3.1	Object Descriptions .....	5
48	3.1.1	Transport Protocol Selection.....	5
49	3.1.2	Message Representation Description .....	5
50	3.1.3	Message Representation Selection .....	6
51	3.2	Function and Predicate Descriptions.....	7
52	3.2.1	Transport Selection .....	7
53	3.2.2	Message Encoding Selection.....	7
54	3.2.3	Open Communication Channel .....	7
55	3.2.4	Close Communication Channel.....	8
56	3.2.5	Activate a Message Transport Protocol .....	8
57	3.2.6	Deactivate a Message Transport Protocol.....	8
58	3.2.7	Select a Message Transport Protocol.....	9
59	3.3	Exceptions.....	9
60	3.3.1	Not Understood Exception Predicates .....	9
61	3.3.2	Refusal Exception Predicates .....	9
62	3.3.3	Failure Exception Propositions.....	10
63	4	Registration with the Directory Facilitator .....	12
64	5	Examples .....	13
65	5.1	Registration with a Directory Facilitator .....	13
66	5.2	Negotiating Message Transport Protocols .....	14
67	5.3	Negotiating Message Representations .....	18
68	6	Paramedic Scenario .....	20
69	6.1	Overview .....	20
70	6.2	Seamless Roaming .....	22
71	6.2.1	Disconnection and Reconnection of an Message Transport Connection .....	22
72	6.2.2	Example Negotiation of a Message Transport Protocol.....	26
73	6.2.3	Example Negotiation of a Message Representation .....	28
74	7	References .....	31
75	8	Informative Annex A — ChangeLog.....	32
76	8.1	2001/10/17 - version E by TC Gateways.....	32
77	8.2	2002/09/13 - version F by TC X2S .....	32
78	8.3	2002/11/01 - version G by TC X2S .....	33
79	8.4	2002/12/03 - version H by FIPA Architecture Board .....	33

## 80 **1 Scope**

81 This document is part of the FIPA specifications and deals with agent middleware to support applications in nomadic  
82 environment. The environment of mobile computing is very different compared to today's environment of traditional  
83 distributed systems in many respects. Bandwidth, latency, delay, error rate, interference, interoperability, computing  
84 power, quality of display, among other things may change dramatically as a nomadic end-user moves from one location  
85 to another. All these cause new demands for adaptability of data services.

86  
87 Adaptability to the changes in the environment of nomadic end-users is an important issue. A nomadic end-user  
88 confronted with these circumstances would benefit from having the following functionality provided by the infrastructure:  
89 information about expected performance, agents controlling over the transfer operations, a condition-based control  
90 policy, capability provided by agents to work in a disconnected mode, advanced error recovery methods, and  
91 adaptability.

92  
93 This specification gives an overview of the nomadic application support area and contains informative specifications for:

- 94 • Monitor Agent (MA) functionality, and
- 95 • Control Agent (CA) functionality.

96  
97  
98  
99 In addition, three other FIPA specifications are related to nomadic application support: [FIPA00069], [FIPA00088] and  
100 [FIPA00094].

101

## 102 2 General Analysis

### 103 2.1 Overview

104 The results of current developments in both wireless data communications and mobile computers are being combined  
 105 to facilitate a new trend: *nomadic computing*. Compared to today's traditional distributed systems, the nomadic  
 106 computing environment is very different in many respects. Bandwidth, latency, delay, error rate, quality of display and  
 107 other non-functional parameters may change dramatically when a nomadic end-user moves from one location to  
 108 another and thus from one computing environment to another, for example, from a wire line LAN to a UMTS network.  
 109 The variety of mobile workstations, handheld devices and smart phones, which allow nomadic end-users to access  
 110 Internet services, is increasing rapidly. The capabilities of mobile devices range from very low performance equipment  
 111 (such as PDAs) up to high performance laptop PCs. All these devices create new demands for adaptability of Internet  
 112 services. For example, PDAs cannot display properly high quality images and as nomadic end-users may be charged  
 113 based on the amount of data transmitted over the GPRS-UMTS network, they may have to pay for bits that are totally  
 114 useless to them.

115  
 116 Confronted with these circumstances, the nomadic end-user would benefit from having the following functionality  
 117 provided by the infrastructure: information about expected performance, agent monitoring and controlling the transfer  
 118 operations, and adaptability.

119  
 120 The ability to automatically adjust to changes in a transparent and integrated fashion is essential for *nomadicity*;  
 121 nomadic end-users are usually professionals in areas other than computing. Furthermore, today's mobile computer  
 122 systems are already very complex to use as productivity tools. Thus, nomadic end-users need all the support that a  
 123 FIPA agent-based distributed system can deliver and adaptability to the changes in the environment of nomadic end-  
 124 users is an important issue.

125  
 126 The adaptation of applications to various nomadic computing environments is an important area. There are several  
 127 tasks that agents need to carry out during application adaptation:

- 128  
 129 1. Selection of Message Transport Protocol (MTP) and Message Transport Connection (MTC) to be used for agent  
 130 communication.
- 131  
 132 2. Selection of an ACL and content language representation to be used for agent communication.
- 133  
 134 3. Provision of support for application agents to carry out adaptation of application data, such as still images, video  
 135 and audio, XML, etc. Today's Internet application data (such as multimedia content) are designed with high  
 136 performance desktop PCs and high quality displays in mind. Therefore, the application data is frequently unsuitable  
 137 for nomadic computing using wireless wide-area networks and low performance mobile devices, and hence  
 138 requires modification.
- 139  
 140 4. Communication between agents performing adaptation.

141  
 142 The FIPA Nomadic Application Support specifications define agent middleware to monitor and control an MTP and the  
 143 underlying MTC. In addition, this specification gives examples of the use of the above scenarios.

### 145 2.2 Monitoring and Controlling Quality of Service

146 The functions required to carry out monitoring and controlling for Quality of Service (QoS) can be split into several  
 147 specific tasks:

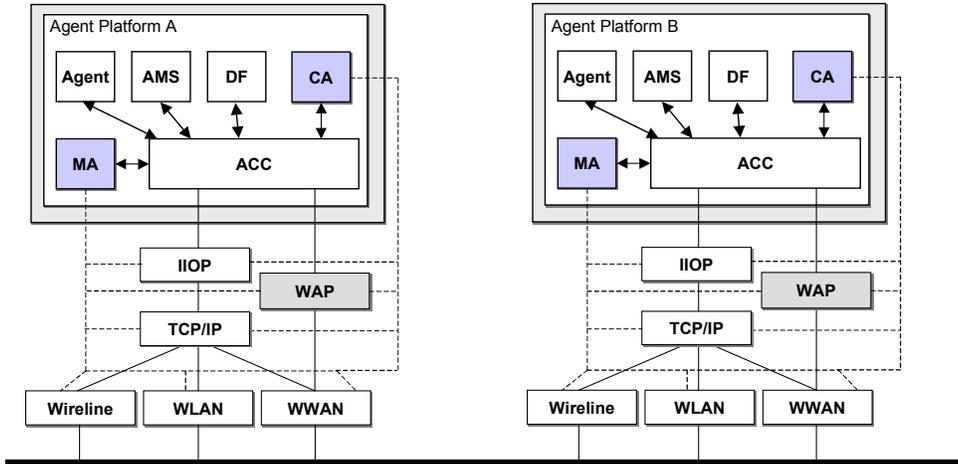
- 148  
 149 1. Observing the QoS of MTPs and MTCs,
- 150  
 151 2. Measuring (if there are no other means to obtain the required information) the QoS of an MTP and MTC,
- 152  
 153 3. Collecting information from the observing and measuring sources,

154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164

4. Analysing the information, and,
5. Controlling an MTC and selecting an MTP.

Based on this division, the agent middleware consists of the following logical agents (see *Figure 1*):

- A MA which carries out tasks 1 through 4, and,
- A CA which carries out task 5.



165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189

**Figure 1:** Reference Model of Agent based Adaptation

The most appropriate configuration of MAs and CAs is that there is at least one pair in each AP involving adaptation. The MA may measure the actual QoS of an MTC, if the network running an MTC does not provide users with required performance data<sup>1</sup>.

An MA may:

- Consist of network-service-specific components that collect raw performance data at fixed intervals,
- Provide a repository for the measurement data collected,
- Perform first level analysis of the collected data, and,
- Send the results of the analysis to CA, if requested to do so.

A CA may:

- Manage (establish, close, suspend, activate, etc.) an MTC<sup>2</sup>.

In some cases there is a need for MAs and CAs in heterogeneous APs to communicate with each other; therefore, interaction protocols and ontologies to achieve this are specified in this document.

<sup>1</sup> The way this actual measurement is performed is not a subject of standardisation within FIPA.

<sup>2</sup> The way that management actions are executed is not a subject of standardisation within FIPA.

**2.3 Negotiation of Message Transport Requirements**

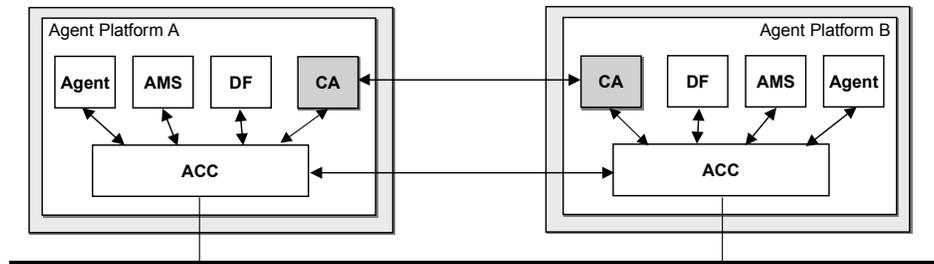
There are several mechanisms that can determine the MTP, message representation and content language to use between communicating entities:

- Communicating entities know a peer entity’s preferences beforehand and use them.
- The activating entity tries to use a method and if the peer entity is not capable of using the suggested method, then the activating entity may try another one (and so on).
- The communicating entities negotiate about a method to be used.

**2.3.1 Negotiation about Message Transport Protocols**

Previous FIPA specifications have implicitly assumed that the MTC is operational all the time (meaning that the MTC has been established before the agent message exchange and that it is reliable). However, this is not always the case within a nomadic environment.

A CA can activate the selection of an MTP or an agent can propose an MTP to a CA and it is the responsibility of the CA to either accept or reject the proposal based on whether it is possible to use the proposed MTP. CAs negotiate with peer CAs to use proposed MTPs which is illustrated in *Figure 2*.



**Figure 2:** Control Agents Negotiating About a Message Transport Protocol

CAs use the `fipa-propose` interaction protocol [FIPA00036] and the `use` action to negotiate about an MTP. An example negotiation is given in Section 5.2.

**2.3.2 Negotiation about Message Representation**

In the environment of nomadic applications, it may be necessary to switch from one ACL representation to another; for example, when a mobile host roams from a wire line network to a wireless network. Application agents may use the `fipa-propose` interaction protocol and the `use` action to negotiate about the representation of ACL. Examples of this negotiation are given in Section 5.3.

## 223 3 Nomadic Application Support Ontology

### 224 3.1 Object Descriptions

225 This section describes a set of frames, that represent the classes of objects in the domain of discourse within the  
 226 framework of the `fipa-nas` ontology. The `fipa-nas` ontology extends the `fipa-qos` ontology defined in  
 227 [FIPA00094].

228  
 229 The following terms are used to describe the objects of the domain:

- 231 • **Frame.** This is the mandatory name of this entity that must be used to represent each instance of this class.
- 232
- 233 • **Ontology.** This is the name of the ontology, whose domain of discourse includes the parameters described in the  
 234 table.
- 235
- 236 • **Parameter.** This is the mandatory name of a parameter of this frame.
- 237
- 238 • **Description.** This is a natural language description of the semantics of each parameter.
- 239
- 240 • **Presence.** This indicates whether each parameter is mandatory or optional.
- 241
- 242 • **Type.** This is the type of the values of the parameter: Integer, Word, String, URL, Term, Set or Sequence.
- 243
- 244 • **Reserved Values.** This is a list of FIPA-defined constants that can assume values for this parameter.
- 245

#### 246 3.1.1 Transport Protocol Selection

247 This type of object represents a selection of transport protocol.

248

Frame Ontology	transports fipa-nas	Parameter	Description	Presence	Type	Reserved Values
send	A list of transport protocols supported for sending messages.	Mandatory	Sequence of transport-protocol <sup>3</sup>			
recv	A list of transport protocols supported for receiving messages.	Mandatory	Sequence of transport-protocol			

249

#### 250 3.1.2 Message Representation Description

251 This type of object represents an ACL message representation.

252

Frame Ontology	msg-representation fipa-nas	Parameter	Description	Presence	Type	Reserved Values
name	The name of the message representation.	Mandatory	word			
options	A list of parameters for the message representation.	Optional	Set of property <sup>4</sup>			

<sup>3</sup> See [FIPA00094].

<sup>4</sup> See [FIPA00023].

253

254 **3.1.3 Message Representation Selection**

255 This type of object represents a selection of message representations.

256

<b>Frame</b>	msg-encoding			
<b>Ontology</b>	fipa-nas			
<b>Parameter</b>	<b>Description</b>	<b>Presence</b>	<b>Type</b>	<b>Reserved Values</b>
send	A list of message representations supported for sending messages.	Mandatory	Sequence of msg-representation	
recv	A list of message representations supported for receiving messages.	Mandatory	Sequence of msg-representation	

257

## 258 3.2 Function and Predicate Descriptions

259 The following tables define usage and semantics of the functions and the predicates that are part of the `fipa-nas`  
 260 ontology.

261 The following terms are used to describe the functions of the `fipa-nas` domain:  
 262

- 264 • **Function.** This is the symbol that identifies the function in the ontology.
- 265
- 266 • **Predicate.** This is the symbol that identifies the predicate in the ontology.
- 267
- 268 • **Ontology.** This is the name of the ontology, whose domain of discourse includes the function or the predicate  
 269 described in the table.
- 270
- 271 • **Supported by.** This is the type of agent that supports this function or predicate.
- 272
- 273 • **Description.** This is a natural language description of the semantics of the function or the predicate.  
 274
- 275 • **Domain.** This indicates the domain over which the function predicate is defined. The arguments passed to the  
 276 function or predicate must belong to the set identified by the domain.
- 277
- 278 • **Range.** This indicates the range to which the function maps the symbols of the domain. The result of the function is  
 279 a symbol belonging to the set identified by the range.
- 280
- 281 • **Arity.** This indicates the number of arguments that a function or a predicate takes. If a function or a predicate can  
 282 take an arbitrary number of arguments, then its arity is undefined.  
 283

### 284 3.2.1 Transport Selection

<b>Predicate</b>	<code>transport-selection</code>
<b>Ontology</b>	<code>fipa-nas</code>
<b>Supported by</b>	CA
<b>Description</b>	An agent specifies the transport protocols that it is willing to use. The predicate is true, when the values of the <code>transports</code> parameter contain the transport protocol descriptions that the agent is willing to use. Otherwise, the predicate is false
<b>Domain</b>	<code>transports</code>
<b>Arity</b>	1

285

### 286 3.2.2 Message Encoding Selection

<b>Predicate</b>	<code>msg-encoding-selection</code>
<b>Ontology</b>	<code>fipa-nas</code>
<b>Supported by</b>	CA
<b>Description</b>	An agent specifies the message encoding choices that it is willing to use. The predicate is true, when the values of the <code>msg-encoding</code> parameter contain the message encoding choices that the agent is willing to use. Otherwise, the predicate is false
<b>Domain</b>	<code>msg-encoding</code>
<b>Arity</b>	1

287

288

### 289 3.2.3 Open Communication Channel

<b>Function</b>	<code>open-comm-channel</code>
-----------------	--------------------------------

<b>Ontology</b>	fipa-nas
<b>Supported by</b>	CA
<b>Description</b>	An agent can request that a CA open a communication channel. The communication channel description should contain enough information for a CA to be able to choose the right communication channel, that is, either the <code>name</code> parameter or the <code>target-addr</code> parameter must be present. The agent may also supply additional communication channel information by using the <code>options</code> parameter.
<b>Domain</b>	comm-channel
<b>Range</b>	The execution of this function results in a change of the state, but it has no explicit result. Therefore there is no range set.
<b>Arity</b>	1

290

291 **3.2.4 Close Communication Channel**

<b>Function</b>	close-comm-channel
<b>Ontology</b>	fipa-nas
<b>Supported by</b>	CA
<b>Description</b>	An agent can request that a CA close a communication channel. The communication channel description should contain enough information for a CA to be able to choose the right communication channel, that is, either the <code>name</code> parameter or the <code>target-addr</code> parameter must be present.
<b>Domain</b>	comm-channel
<b>Range</b>	The execution of this function results in a change of the state, but it has no explicit result. Therefore there is no range set.
<b>Arity</b>	1

292

293 **3.2.5 Activate a Message Transport Protocol**

<b>Function</b>	activate
<b>Ontology</b>	fipa-nas
<b>Supported by</b>	CA
<b>Description</b>	An agent can request that a CA activate a Message Transport Protocol (MTP). The transport protocol description should contain enough information to allow the CA to identify the correct transport protocol. Additionally, the agent may supply address information to where the transport protocol connection should be opened. It is possible to give the address of the gateway and/or the address of the destination AP. If the action is successful, the CA will return the object description of activated MTP.
<b>Domain</b>	Sequence of <code>transport-protocol</code>
<b>Range</b>	<code>transport-protocol</code>
<b>Arity</b>	1

294

295 **3.2.6 Deactivate a Message Transport Protocol**

<b>Function</b>	deactivate
<b>Ontology</b>	fipa-nas
<b>Supported by</b>	CA
<b>Description</b>	An agent can request that a CA deactivate an MTP.
<b>Domain</b>	<code>transport-protocol</code>
<b>Range</b>	The execution of this function results in a change of the state, but it has no explicit result. Therefore there is no range set.
<b>Arity</b>	1

296

297 **3.2.7 Select a Message Transport Protocol**

<b>Function</b>	use
<b>Ontology</b>	fipa-nas
<b>Supported by</b>	CA
<b>Description</b>	An CA can request another CA to select an MTP or message encoding for use between Agent Communication Channels (ACCs). The requesting CA shall provide enough information to establish a working MTP connection or message encoding. The direction of communication (either send, receive or both) and the list of choices must be present. The list of choices is an ordered list where the highest priority is the first item and the lowest priority is the last item in the list. The receiving CA shall select at most one choice for the proposed direction of communication (either send, receive or both)
<b>Domain</b>	transports / <sup>5</sup> msg-encoding
<b>Range</b>	transport-selection / <sup>6</sup> msg-encoding-selection
<b>Arity</b>	1

298

299 **3.3 Exceptions**

300 The exceptions for the fipa-nas ontology follow the same form and rules as specified in [FIPA00023].

301

302 **3.3.1 Not Understood Exception Predicates**

<b>Communicative Act Ontology</b>	not-understood fipa-nas	
<b>Predicate Symbol</b>	<b>Arguments</b>	<b>Description</b>
unsupported-act	string	The receiving agent does not support the specific communicative act; the string identifies the unsupported communicative act.
unexpected-act	string	The receiving agent supports the specified communicative act, but it is out of context; the string identifies the unexpected communicative act.
unsupported-value	string	The receiving agent does not support the value of a message parameter; the string identifies the message parameter name.
unrecognised-value	string	The receiving agent cannot recognise the value of a message parameter; the string identifies the message parameter name.

303

304 **3.3.2 Refusal Exception Predicates**

<b>Communicative Act Ontology</b>	refuse fipa-nas	
<b>Predicate symbol</b>	<b>Arguments</b>	<b>Description</b>
unauthorised		The sending agent is not authorised to perform the function.
unsupported-function	string	The receiving agent does not support the function; the string identifies the unsupported function name.
missing-argument	string	A mandatory function argument is missing; the string identifies the missing function argument name.

<sup>5</sup> Where '/' is "exclusive or".<sup>6</sup> Where '/' is "exclusive or".

unexpected-argument	string	A mandatory function argument is present which is not required; the string identifies the function argument that is not expected.
unexpected-argument-count		The number of function arguments is incorrect.
missing-parameter	string string	A mandatory parameter is missing; the first string represents the object name and the second string represents the missing parameter name.
unexpected-parameter	string string	The receiving agent does not support the parameter; the first string represents the function name and the second string represents the unsupported parameter name.
unrecognised-parameter-value	string string	The receiving agent cannot recognise the value of a parameter; the first string represents the object name and the second string represents the parameter name of the unrecognised parameter value.
already-open	string	The specified communication channel is already open; the string identifies the communication channel.
not-open	string	The specified communication channel is not open; the string identifies the communication channel.
already-activated	string	The specified transport protocol is already activated; the string identifies the transport protocol.
not-active	string	The specified transport protocol is not active; the string identifies the transport protocol.
unrecognised-comm-channel	string	The specified communication channel is not recognised; the string identifies the communication channel.
unsupported-protocol	string	The specified transport protocol is not supported; the string identifies the transport protocol.

305

306

**3.3.3 Failure Exception Propositions**

<b>Communicative Act Ontology</b>	failure fipa-nas	
<b>Predicate symbol</b>	<b>Arguments</b>	<b>Description</b>
internal-error	string	An internal error occurred; the string identifies the internal error.
open-failed	string	The opening of a communication channel failed; the string identifies the failure reason.
transient-failed	string	The opening/closing of a communication channel or the activation/deactivation of a transport protocol failed; the string identifies the failure reason.
close-failed	string	The closing of a communication channel failed; the string identifies the failure reason.
activation-failed	string	The activation of a transport protocol failed; the string identifies the failure reason.
deactivation-failed	string	The deactivation of a transport protocol failed; the string identifies the failure reason.



## 308 4 Registration with the Directory Facilitator

309 In order for a CA and MA to advertise its willingness to provide its services to an agent domain, it must register with a  
 310 DF (as described in [FIPA00023]. As part of this registration process, the following of constant values are introduced  
 311 that universally identify the services the agent provides:

- 312
- 313 • The name parameter in service-description frame of a CA must be declared as a constant `fipa-mts-`  
 314 `control`.
- 315
- 316 • The type parameter in service-description frame of a CA must be declared as a constant `fipa-ca`.
- 317
- 318 • The ontology parameter in service-description frame of a CA should be declared as a constant `fipa-`  
 319 `nas`.
- 320
- 321 • The type parameter in service-description frame of a MA must be declared as a constant `fipa-mts-`  
 322 `monitor`.
- 323
- 324 • The type parameter in service-description frame of a MA must be declared as a constant `fipa-ma`.
- 325
- 326 • The ontology parameter in service-description frame of a MA should be declared as a constant `fipa-`  
 327 `qos`.
- 328

329 Below is given an example content of a `df-agent-description` frame which provides both MA and CA functionality:

```
330
331 (df-agent-description
332   :name
333     (agent-identifier
334       :name monitor&control_agent@iiop://foo.com/acc
335       :addresses (sequence iiop://foo.com/acc))
336   :protocols (set fipa-request fipa-propose)
337   :ontology (set fipa-nas)
338   :language (set fipa-sl)
339   :services (set
340     (service-description
341       :name fipa-mts-control
342       :type fipa-ca
343       :ontology fipa-nas)
344     (service-description
345       :name fipa-mts-monitor
346       :type fipa-ma
347       :ontology fipa-qos))
348   :ownership (set Sonera))))
349
```

## 350 5 Examples

### 351 5.1 Registration with a Directory Facilitator

#### 352 1. A CA registers with a DF (see [FIPA00023]):

```

353 (request
354   :sender
355     (agent-identifier
356       :name ca@foo.com
357       :addresses (sequence http://foo.com/acc))
358   :receiver (set
359     (agent-identifier
360       :name df@foo.com
361       :addresses (sequence http://foo.com/acc)))
362   :language fipa-sl
363   :protocol fipa-request
364   :ontology fipa-agent-management
365   :content "(
366     (action
367       (agent-identifier
368         :name df@foo.com
369         :addresses (sequence http://foo.com/acc))
370       (register
371         (df-agent-description
372           :name
373             (agent-identifier
374               :name ca@foo.com
375               :addresses (sequence http://foo.com/acc))
376           :services (set
377             (service-description
378               :name fipa-mts-control
379               :type fipa-ca
380               :ontology (set fipa-nas))))))))))")
381

```

#### 382 2. An MA registers with a DF.

```

383 (request
384   :sender
385     (agent-identifier
386       :name ma@foo.com
387       :addresses (sequence http://foo.com/acc))
388   :receiver (set
389     (agent-identifier
390       :name df@foo.com
391       :addresses (sequence http://foo.com/acc)))
392   :language fipa-sl
393   :protocol fipa-request
394   :ontology fipa-agent-management
395   :content "(
396     (action
397       (agent-identifier
398         :name df@foo.com
399         :addresses (sequence http://foo.com/acc))
400       (register
401         (df-agent-description
402           :name
403             (agent-identifier
404               :name ma@foo.com
405               :addresses (sequence http://foo.com/acc))
406           :services (set
407

```

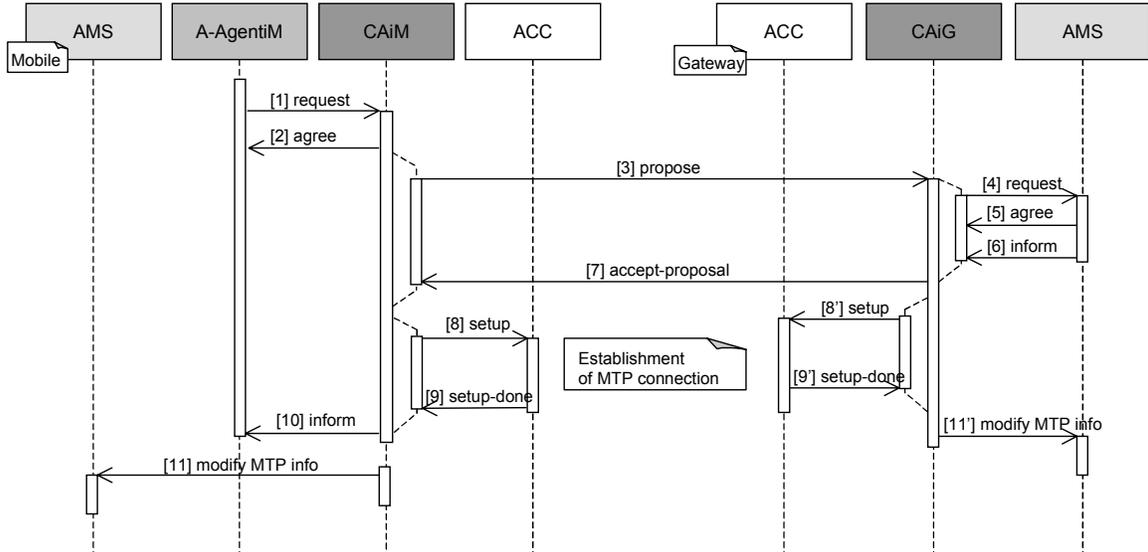
```

409         (service-description
410           :name fipa-mts-monitor
411           :type fipa-ma
412           :ontology (set fipa-nas)))))))))")
413

```

## 5.2 Negotiating Message Transport Protocols

This example shows a scenario, where an application agent requests the use of either the WAP MTP [FIPA00076] or a proprietary MTP (for example, x.uh.mdcP). The message flow of a successful negotiation is illustrated in *Figure 3*.



**Figure 3:** Flow of Message Transport Protocol Negotiation

1. Message 1 request: An application agent issues a request to the CA to activate either the fipa.mts.mtp.wap.std or x.uh.mdcP MTPs.

```

425 (request
426   :sender
427     (agent-identifier
428       :name A-AgentiM@mobile.com7)
429   :receiver (set
430     (agent-identifier
431       :name CaiM@mobile.com))
432   :ontology fipa-nas
433   :language fipa-sl
434   :protocol fipa-request
435   :content "(
436     (action
437       (agent-identifier
438         :name CAiM@mobile.com)
439       (activate (sequence
440         (transport-protocol
441           :name x.uh.mdcP)
442         (transport-protocol
443           :name fipa.mts.mtp.wap.std
444           :dest-addr wap://gateway.com:1234/acc))))))")
445

```

<sup>7</sup> In all of the examples in this specification, the suffix of iM in an agent's name represents a mobile host, that is, an agent that is located on a mobile AP. Similarly, the suffix iG represents a gateway host and the suffix iF represents a fixed network host.

- 446 2. Message 2 agree: The CA agrees to activate an MTP. The decision to agree or disagree to activate an MTP might  
 447 be based on the internal state of the CA (that is, the CA knows whether a requested MTP can be activated or not)  
 448 or the CA might ask for an AP description from an AMS.

```

449 (agree
450   :sender
451     (agent-identifier
452       :name CAiM@mobile.com)
453   :receiver (set
454     (agent-identifier
455       :name A-AgentiM@mobile.com))
456 :ontology fipa-nas
457 :language fipa-sl
458 :protocol fipa-request
459 :content "(
460   (action
461     (agent-identifier
462       :name CAiM@mobile.com)
463     (activate (sequence
464       (transport-protocol
465         :name x.uh.mdcp)
466       (transport-protocol
467         :name fipa.mts.mtp.wap.std
468         :dest-addr wap://gateway.com:1234/acc))))
469   true))"
470 )
471 
```

- 472 3. Message 3 propose: The CA in the mobile host proposes to its peer CA in the gateway host that either the  
 473 fipa.mts.mtp.wap.std or x.uh.mdcp MTPs should be used in communication between the APs.

```

474 <?xml version="1.0"?>8
475 <envelope>
476   <params index="1">
477     <to>
478       <agent-identifier>
479         <name>CAiG@gateway.com</name>
480       </agent-identifier>
481     </to>
482     <from>
483       <agent-identifier>
484         <name>CAiM@mobile.com</name>
485       </agent-identifier>
486     </from>
487     <acl-representation>fipa.acl.rep.string.std</acl-representation>
488     <date>20000606T100900000</date>
489   </params>
490 </envelope>
491 (propose
492   :sender
493     (agent-identifier
494       :name CAiM@mobile.com)
495   :receiver (set
496     (agent-identifier
497       :name CAiG@gateway.com)
498     (agent-identifier
499       :name CAiM@mobile.com))
500 )
501 
```

<sup>8</sup> In most of the examples, the envelope part has been omitted for clarity.

```

506         :name CAiG@gateway.com))
507 :ontology fipa-nas
508 :language fipa-sl
509 :protocol fipa-propose
510 :content "(
511   (action
512     (agent-identifier
513       :name CAiM@mobile.com)
514     (use
515       (transports
516         :send (sequence
517           (transport-protocol
518             :name x.uh.mdcp)
519           (transport-protocol
520             :name fipa.mts.mtp.wap.std))
521         :recv (sequence
522           (transport-protocol
523             :name x.uh.mdcp)
524           (transport-protocol
525             :name fipa.mts.mtp.wap.std))))))
526   true) ")
527

```

4. Message 4 request, message 5 agree and message 6 inform: The CA in the gateway host requests the AP description from the local AMS (see [FIPA00023]) to determine whether the `x.uh.mdcp` or `fipa.mts.mtp.wap.std` MTPs are supported. The AMS informs the CA that both MTPs are supported and the CA decides to use `fipa.mts.mtp.wap.std` MTP based on the current QoS requirements of the MTC.

```

533 (request
534   :sender
535     (agent-identifier
536       :name CAiG@gateway.com)
537   :receiver (set
538     (agent-identifier
539       :name ams@gateway.com))
540   :ontology fipa-agent-management
541   :language fipa-sl
542   :protocol fipa-request
543   :content "(
544     (action
545       (agent-identifier
546         :name ams@gateway.com)
547       get-description)) ")
548
549 (agree
550   :sender
551     (agent-identifier
552       :name ams@gateway.com)
553   :receiver (set
554     (agent-identifier
555       :name CAiG@gateway.com))
556   :ontology fipa-agent-management
557   :language fipa-sl
558   :protocol fipa-request
559   :content "(
560     (action
561       (agent-identifier
562         :name ams@gateway.com)
563       get-description)
564     true) ")
565
566 (inform
567   :sender
568     (agent-identifier

```

```

569         :name ams@gateway.com
570         :addresses (sequence http://gateway.com/acc))
571 :receiver (set
572   (agent-identifier
573     :name CAiG@gateway.com
574     :addresses (sequence http://gateway.com/acc)))
575 :ontology fipa-agent-management
576 :language fipa-sl
577 :protocol fipa-request
578 :content "(
579   (result
580     (action
581       (agent-identifier :name ams@gateway.com)
582       get-description)
583     (ap-description
584       :name sonera-platform
585       :transport-profile
586       (ap-transport-description
587         :available-mtps
588           (set
589             (mtp-description
590               :profile fipa.profile.mts.alpha
591               :mtp-name fipa.mts.mtp.iiop.std
592               :addresses (sequence iiop://gateway.com/acc))
593             (mtp-description
594               :profile fipa.profile.mts.beta
595               :mtp-name fipa.mts.mtp.wap.std
596               :addresses (sequence wap://gateway.com:1234/acc))
597             (mtp-description
598               :profile x.uh.profile
599               :mtp-name x.uh.mdcp
600               :addresses (set mdcp://gateway.com/acc)))))))")

```

5. Message 7 accept-proposal: The CA in the gateway host accepts the proposal to use the fipa.mts.mtp.wap.std MTP and sends the response to the CA in the mobile host informing it about the preferred MTP.

```

606 (accept-proposal
607   :sender
608     (agent-identifier
609       :name CAiG@gateway.com)
610   :receiver (set
611     (agent-identifier
612       :name CAiM@mobile.com))
613   :ontology fipa-nas
614   :language fipa-sl
615   :protocol fipa-propose
616   :content "(
617     (action
618       (agent-identifier
619         :name CAiM@mobile.com)
620       (use
621         (transports
622           :send (sequence
623             (transport-protocol
624               :name x.uh.mdcp)
625             (transport-protocol
626               :name fipa.mts.mtp.wap.std))
627         :recv (sequence
628           (transport-protocol
629             :name x.uh.mdcp)
630           (transport-protocol
631             :name fipa.mts.mtp.wap.std))))))

```

```

632     (transport-selection
633       (transports
634         :send (sequence
635           (transport-protocol
636             :name fipa.mts.mtp.wap.std))
637         :recv (sequence
638           (transport-protocol
639             :name fipa.mts.mtp.wap.std))))))")
640

```

- 641 6. Messages 8 and 8' setup: The CAs request their respective ACCs to setup the `fipa.mts.mtp.wap.std` MTP. This is an implementation issue.
- 642
- 643
- 644 7. Message 9 and 9' setup-done: The ACCs inform their respective CAs that the `fipa.mts.mtp.wap.std` MTP has been established between the mobile host and the gateway host.
- 645
- 646
- 647 8. Message 10 inform: The CA informs the application agent that the MTC is established.

```

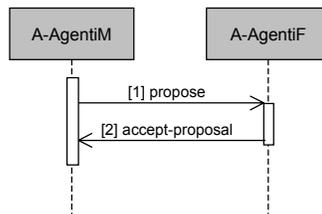
648 (inform
649   :sender
650     (agent-identifier
651       :name CAiM@mobile.com)
652   :receiver (set
653     (agent-identifier
654       :name A-AgentiM@mobile.com))
655   :ontology fipa-nas
656   :language fipa-sl
657   :protocol fipa-request
658   :content "(
659     (result
660       (action
661         (agent-identifier
662           :name CaiM@mobile.com)
663         (activate (sequence
664           (transport-protocol
665             :name x.uh.mdcp)
666           (transport-protocol
667             :name fipa.mts.mtp.wap.std
668             :dest-addr wap://gateway.com:1234/acc))))
669     (transport-protocol
670       :name fipa.mts.mtp.wap.std)))")
671

```

- 672
- 673 9. Message 11 and 11' set-description: CAiM (/CAiG) modifies the AP description to show that the `fipa.mts.mtp.wap.std` is now active.
- 674
- 675

### 5.3 Negotiating Message Representations

This example shows a scenario where an application agent in a mobile host proposes to its peer application agent in a fixed host the use of the `fipa.acl.rep.bitefficient.std` representation of ACL [FIPA00069] for their communication. The message flow is illustrated in *Figure 4*.



**Figure 4:** Flow of Message Representation Negotiation

681  
682  
683  
684

685 1. Message 1 propose: The agent in the mobile host proposes the use of the `fipa.acl.rep.bitefficient.std`  
 686 representation of ACL.

```
687
688 (propose
689   :sender
690     (agent-identifier
691       :name A-AgentiM@mobile.com)
692   :receiver (set
693     (agent-identifier
694       :name A-AgentiF@fixed.com))
695   :ontology fipa-nas
696   :language fipa-sl
697   :protocol fipa-propose
698   :content "(
699     (action
700       (agent-identifier
701         :name A-AgentiM@mobile.com)
702       (use
703         (msg-rep-selection
704           :send (sequence
705             (msg-representation
706               :name fipa.acl.rep.bitefficient.std))
707           :rcv (sequence
708             (msg-representation
709               :name fipa.acl.rep.bitefficient.std))))))
710     true) ")
```

711  
 712 2. Message 2 accept-proposal: The agent in the fixed host accepts the proposal.

```
713
714 (accept-proposal
715   :sender
716     (agent-identifier
717       :name A-AgentiF@fixed.com)
718   :receiver (set
719     (agent-identifier
720       :name A-AgentiM@mobile.com))
721   :ontology fipa-nas
722   :language fipa-sl
723   :protocol fipa-propose
724   :content "(
725     (action
726       (agent-identifier
727         :name A-AgentiM@mobile.com)
728       (use
729         (msg-encoding
730           :send (sequence
731             (msg-representation :name fipa.acl.rep.bitefficient.std))
732           :rcv (sequence
733             (msg-representation :name fipa.acl.rep.bitefficient.std))))))
734     (msg-encoding-selection
735       (msg-encoding
736         :send (sequence
737           (msg-representation :name fipa.acl.rep.bitefficient.std))
738         :rcv
739           (sequence
740             (msg-representation :name fipa.acl.rep.bitefficient.std))))))")
741
```

## 742 6 Paramedic Scenario

743 This section illustrates some of the important issues of nomadic application support, using a paramedic application as  
744 an example.  
745

### 746 6.1 Overview

747 A paramedic team has several working environments:

- 748
- 749 • An emergency dispatch centre, which is covered by the hospital ATM network,
- 750
- 751 • A geographical area, which is wireless wide-area network (for example, GPRS), and,
- 752
- 753 • One or more hospitals, which are provided with a wireless local-area network.
- 754

755 When in transit, the paramedic computers are attached to docking stations residing in ambulances. At the dispatch  
756 centre, the docking stations are connected to the ATM network. The paramedic application comprises the following  
757 services:

- 758
- 759 • Retrieval of a patient's personal information, such as name, address, phone, and relatives,
- 760
- 761 • Retrieval of the patient's medical histories,
- 762
- 763 • Support for paramedic workers, and,
- 764
- 765 • Informing the hospital receiving the patient about the patient's current injury or illness and medical care given so far.
- 766

767 There are several application agents: Paramedic Support Agents (PSAs) working in the paramedic computers,  
768 Dispatching Support Agent (DSA) working at the dispatch centre system, and the Hospital First Aid Support Agent  
769 (HFASA) working at the hospital system.

770

771 The dispatch centre receives a call regarding a man who has severe chest pain; the symptom of an acute myocardial  
772 infarct. The caller identifies the man and gives his personal identification number to the dispatcher. The dispatcher  
773 alerts the paramedic team and informs the DSA about the address where the patient is located and his personal  
774 identification number. The DSA simultaneously informs the PSA about the address of the attack (and possibly some  
775 additional information about the environment of the heart attack) and queries the patient's medical history. Since the  
776 results of the query to a local hospital are received before the paramedic unit is dispatched, the DSA (in co-operation  
777 with the PSA) begins to load the patient's personal information and medical history into the paramedic computers. The  
778 medical history includes several items of text-based information. The transmission time to load the information via the  
779 ATM network to the paramedic computers (which are currently docked at the dispatch centre) is less than a second.  
780 Before the ambulance leaves the dispatch centre, the docking station is detached from the ATM network and is  
781 connected to the wireless wide-area network.

782

783 While the ambulance is approaching the location of the incident, the DSA receives more relevant results of the query of  
784 the medical histories such as the latest heart operation of the patient. The medical history comprises several parts of  
785 textual information and several images and the DSA begins loading the information. As the loading takes place when  
786 the ambulance is in motion, the DSA finds out that the quality of transport service is too low for loading some textual  
787 parts and any of the images of the medical history. It would take at least 40 minutes to download the images. Therefore,  
788 the DSA informs the PSA that images are not required for the paramedic unit. During downloading, the ambulance  
789 drives into a tunnel that causes the wireless link to be disconnected. After the tunnel, a CA re-establishes the  
790 connection and downloading continues.

791

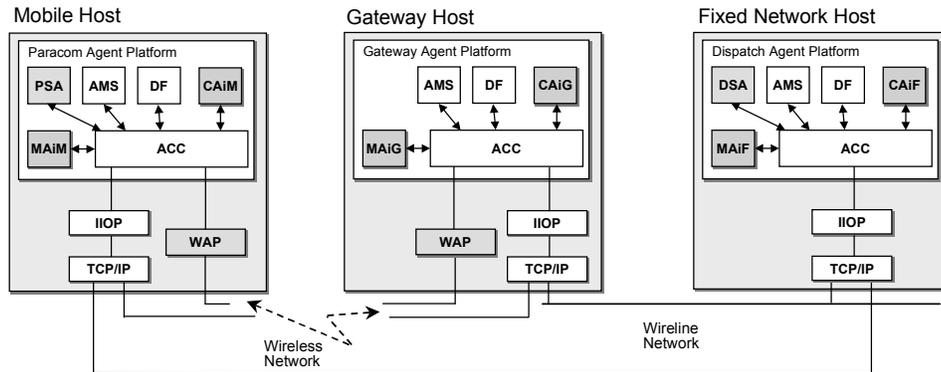
792 At the scene, the ambulance is stationary and the quality of transmission service increases to a level at which the DSA  
793 is able to load the most relevant images (the ECGs) using an efficient compression method which is negotiated

794 between the DSA and the PSA to the paramedic computer. The paramedic team detaches the computers from the  
 795 docking station and carries them to the patient.  
 796

797 The paramedic team realises that they need the assistance of a medical expert located at the university hospital to  
 798 stabilise the patient's condition. Therefore, they attach electrodes to the patient and the PSA starts transmitting the data  
 799 of measurement such as SpO2 (oxygen saturation), cardiac rhythm, ECG, end tidal CO2 and temperature to the  
 800 hospital. After successfully stabilising the patient's condition, the paramedic team moves the patient to the ambulance  
 801 and sets off for the hospital. As the quality of the transport service decreases because of the motion, the PSA finds out  
 802 that not all the on-going measurement data can be transmitted on-line to the hospital. Therefore, the PSA decides to  
 803 transmit the most relevant data (SpO2 and cardiac rhythm). The PSA stores the rest of the data (ECG, end tidal CO2  
 804 and temperature) into a cache of the paramedic computer.  
 805

806 After the ambulance arrives at the hospital, the patient is transferred immediately to an operating room. Simultaneously,  
 807 the paramedic team connects their paramedic computer to the wireless LAN of the hospital and the PSA transmits (in  
 808 co-operation with the HFASA) all the measurement data to the hospital's system. A surgeon retrieves and analyses the  
 809 measurement data before surgery.  
 810

811 This example illustrates a future agent-based distributed system that offers its services at the best obtainable QoS in a  
 812 wide variety of environments. A possible agent architecture is illustrated in *Figure 5* which refers to three separate APs:  
 813 *Dispatch*, *Gateway* and *Paracom*. In addition, there are several hospital APs which are not illustrated.  
 814



815  
 816  
 817 **Figure 5: Paramedic Scenario Architecture**  
 818

819 The agents in the scenario are:

- 820
- 821 • *MAiM*, *MAiG* and *MAiF* are MAs which monitor the quality of the communication service, and,
  - 822
  - 823 • *CAiM*, *CAiG* and *CAiF* are CAs which manage the establishment, teardown, suspension, activation, etc. of the  
 824 connection between the PAs. The MA informs application agents about the status and changes of the network  
 825 services.  
 826

827 When the mobile host is connected either to the ATM network or to the wireless LAN, the *fipa.mts.mtp.iioop.std*  
 828 MTP is used directly between the *Paracom* AP and the *Dispatch* AP. When the mobile host is connected to the wireless  
 829 WAN, all agent message communication takes place through the gateway host. The *fipa.mts.mtp.wap.std* MTP is  
 830 primarily used between the *Paracom* AP and the *Gateway* AP. The *fipa.mts.mtp.iioop.std* MTP is used between  
 831 the *Gateway* AP and the *Dispatch* AP.  
 832

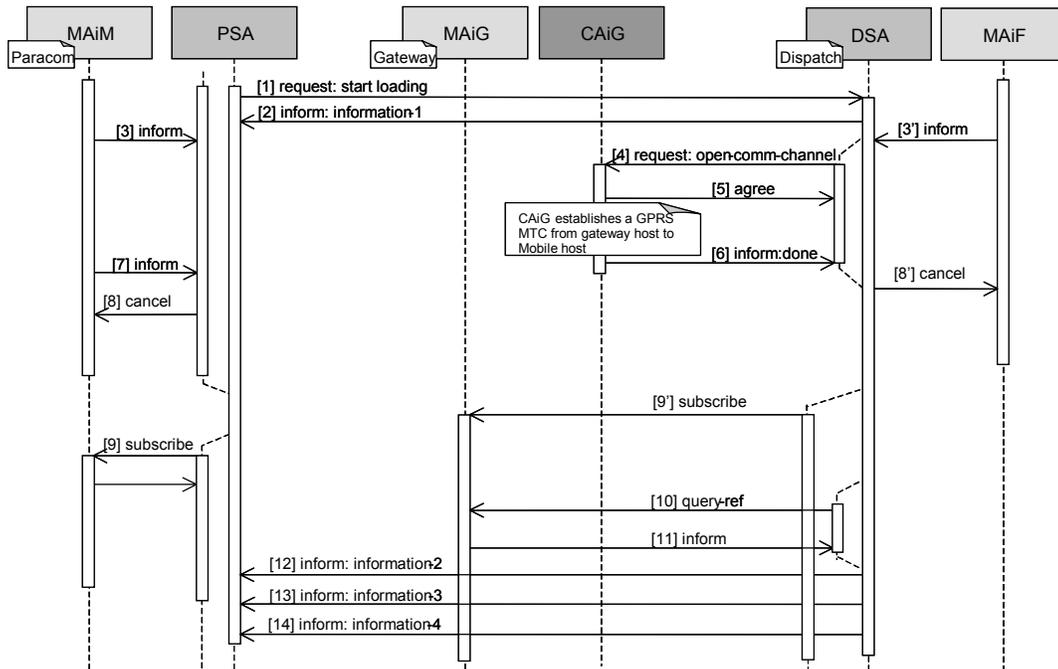
833 **6.2 Seamless Roaming**

834 The Seamless Roaming scenario describes the process, when the paramedic computer roams from the ATM network  
 835 to the UMTS network. The scenario is split into following events:  
 836

- 837 • Disconnection and reconnection of MTCs,
- 838
- 839 • Negotiation of MTPs, and,
- 840
- 841 • Negotiation of message representations.
- 842

843 **6.2.1 Disconnection and Reconnection of an Message Transport Connection**

844 The message exchange between the agents is illustrated in *Figure 6*.  
 845



846  
 847

**Figure 6:** Disconnection and Reconnection of a Message Transport Connection

- 850 1. Message 1 *request*: The PSA starts loading data from the DSA by sending a *request* message. This message is  
 851 application specific and thus not shown here.
- 852
- 853 2. Message 2 *inform*: The DSA starts sending information by first sending an *inform* message.
- 854
- 855 3. Messages 3 and 3' *inform*: MAiM (/MAiF) informs the PSA (/DSA) that the ATM connection has broken.

```

856 (inform
857   :sender
858     (agent-identifier
859       :name MAiM@paracom.com)
860   :receiver (set
861     (agent-identifier
862       :name PSA@paracom.com))
863   :ontology fipa-nas
864   :language fipa-sl
865   :protocol fipa-subscribe
866   :conversation-id subscription-3105
867
```

```

868     :content "(
869         (qos-information
870         (comm-channel
871         :name ATM
872         :target-addr iiop://dispatch.com/acc)
873         (qos
874         :status disconnected))))")
875

```

4. Message 4 request: The DSA requests CAiG to open a wireless wide-area MTC.

```

877
878 (request
879   :sender
880     (agent-identifier
881     :name DSA@dispatch.com)
882   :receiver (set
883     (agent-identifier
884     :name CAiG@gateway.com))
885   :ontology fipa-nas
886   :language fipa-sl
887   :protocol fipa-request
888   :content "(
889     (action
890     (agent-identifier
891     :name CAiG@gateway.com)
892     (open-comm-channel
893     (comm-channel
894     :name GPRS
895     :target-addr iiop://paramedic.com/acc))))")
896

```

5. Message 5 agree: CAiG agrees that it will try to open the GPRS connection.

```

897
898 (agree
899   :sender
900     (agent-identifier
901     :name CAiG@gateway.com)
902   :receiver (set
903     (agent-identifier
904     :name DSA@dispatch.com))
905   :ontology fipa-nas
906   :language fipa-sl
907   :protocol fipa-request
908   :content "(
909     (action
910     (agent-identifier
911     :name CAiG@gateway.com)
912     (open-comm-channel
913     (comm-channel
914     :name GPRS
915     :target-addr iiop://paramedic.com/acc))))
916   true)")
917

```

Next CAiG establishes a GPRS MTC from the gateway host to the mobile host (his is an implementation issue).

6. Message 6 inform: After successful establishment, CAiG informs the DSA.

```

922 (inform
923   :sender
924     (agent-identifier
925     :name CAiG@gateway.com)
926   :receiver (set
927     (agent-identifier
928     :name DSA@dispatch.com))
929

```

```

930 :ontology fipa-nas
931 :language fipa-sl
932 :protocol fipa-request
933 :content "(
934   (done
935     (action
936       (agent-identifier :name CAiG@gateway.com)
937       (open-comm-channel
938         (comm-channel :name gprs :target-addr iiop://paramedic.com/acc))))))"
939

```

7. Message 7 inform: MAiM informs the PSA that a new MTC has been established

```

941 (inform
942   :sender
943     (agent-identifier
944       :name MAiM@paracom.com)
945   :receiver (set
946     (agent-identifier
947       :name PSA@paracom.com))
948   :ontology fipa-nas
949   :language fipa-sl
950   :protocol fipa-subscribe
951   :conversation-id subscription-3105
952   :content "(
953     (qos-information
954       (comm-channel
955         :name GPRS
956         :target-addr wap://paramedic.com:1234/acc)
957       (qos
958         :status disconnected))))"
959
960

```

8. Message 8 and 8' cancel: The PSA (/DSA) cancels subscription notifications about the changes in the ATM MTC.

```

961 (cancel
962   :sender
963     (agent-identifier
964       :name PSA@paracom.com)
965   :receiver (set
966     (agent-identifier
967       :name MAiM@paracom.com))
968   :ontology fipa-nas
969   :language fipa-sl
970   :protocol fipa-subscribe
971   :content "(
972     (iota ?x
973       (exists ?y
974         (and
975           (qos-matches ?x
976             (qos-information
977               (comm-channel
978                 :name gprs
979                 :target-addr wap://paramedic.com:1234/acc)
980               (qos :status ?y)))
981           (or (= ?y connected) (= ?y disconnected))))))"
982
983
984

```

9. Message 9 and 9' subscribe: The DSA (/PSA) subscribes to MAiG (/MAiM) for notifications about the changes in the GPRS MTC.

```

987 (subscribe
988   :sender
989     (agent-identifier
990       :name DSA@dispatch.com)
991   :receiver (set
992

```

```

993     (agent-identifier
994       :name MAiG@gateway.com))
995 :ontology fipa-nas
996 :language fipa-sl
997 :protocol fipa-request
998 :content "(
999   (iota ?x
1000     (and
1001       (time-constraint (time-type :value every) (time-value :value 10 :unit s))
1002       (qos-matches ?x
1003         (qos-information
1004           (comm-channel
1005             :name gprs
1006             :target-addr iiop://paramedic.comm/acc))))))")

```

10. Message 10 query-ref: The DSA requests current QoS of the GPRS MTC from MAiG.

```

1009 (query-ref
1010   :sender
1011     (agent-identifier
1012       :name DSA@dispatch.com)
1013   :receiver (set
1014     (agent-identifier
1015       :name MAiG@gateway.com))
1016   :ontology fipa-nas
1017   :language fipa-sl
1018   :protocol fipa-query
1019   :content "(
1020     (iota ?x
1021       (qos-information
1022         (comm-channel
1023           :name gprs)
1024         (qos
1025           :throughput ?x))))")

```

11. Message 11 inform: MAiG informs the DSA the current QoS of the GPRS MTC.

```

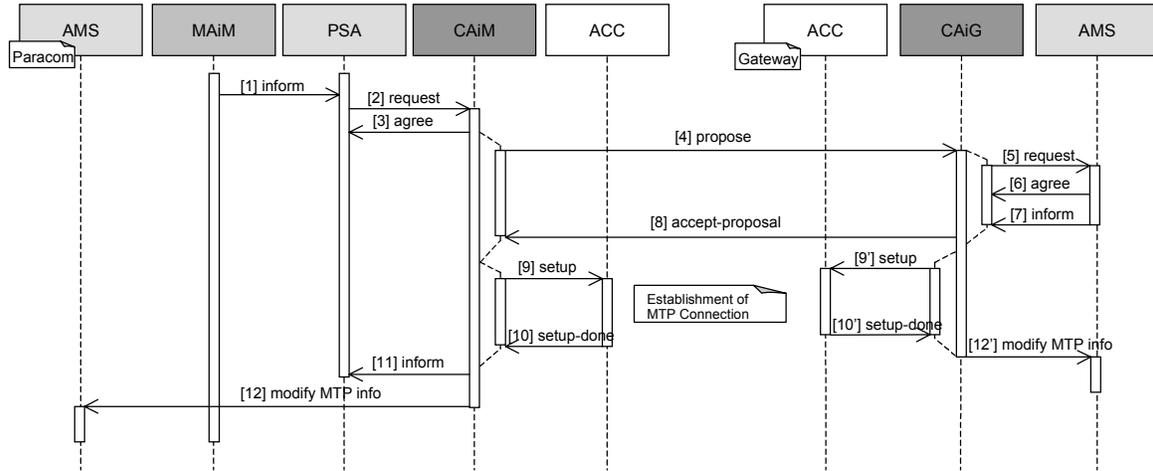
1029 (inform
1030   :sender
1031     (agent-identifier
1032       :name MAiG@gateway.com)
1033   :receiver (set
1034     (agent-identifier
1035       :name DSA@dispatch.com))
1036   :ontology fipa-nas
1037   :language fipa-sl
1038   :protocol fipa-query
1039   :content "(
1040     (= (iota ?x
1041       (qos-information
1042         (comm-channel
1043           :name gprs)
1044         (qos
1045           :throughput ?x)))
1046       (rate-value
1047         :direction outbound
1048         :unit kbits/s
1049         :value 20))))")

```

12. Messages 12, 13 and 14 inform: The DSA sends the rest of the requested information to the PSA.

1054 **6.2.2 Example Negotiation of a Message Transport Protocol**

1055 When the mobile host roams from the ATM network to the GPRS network – after the reconnection – the PSA receives  
 1056 the information from MAiM that the *Paracom* AP is now connected to the GPRS MTC. The PSA reasons that the  
 1057 *fipa.mts.mtp.wap.std* MTP is better in that environment and it requests the CAiM to establish this MTP between  
 1058 ACCiM and ACCiG. Also, CAiM proposes the establishment of this MTP to CAiG, which accepts the proposal, and they  
 1059 command their respective ACCs to set it up. As a last action, both CAiF and CAiG modify the AP descriptions of their  
 1060 APs. The message flow is illustrated in *Figure 7*.  
 1061



1062 **Figure 7: Example Negotiation of a Message Transport Protocol**  
 1063  
 1064  
 1065

- 1066 1. Message 1 *inform*: MAiM informs the PSA that the *Paracom* AP is now connected to the GPRS network.

```

1067 (inform
1068   :sender
1069     (agent-identifier
1070       :name MAiM@paracom.com)
1071   :receiver (set
1072     (agent-identifier
1073       :name PSA@paracom.com))
1074   :ontology fipa-nas
1075   :language fipa-sl
1076   :protocol fipa-subscribe
1077   :conversation-id subscription-3106
1078   :content "(
1079     (qos-information
1080       (comm-channel
1081         :name gprs
1082         :target-addr wap://paramedic.com:1234/acc)
1083       (qos
1084         :status connected)))")
1085
1086
    
```

1087 2. **Message 2 request and message 3 agree:** The PSA requests CAiM to establish the `fipa.mts.mtp.wap.std`  
 1088 MTP between ACCiM and ACCiG.

```
1089 (request
1090   :sender
1091     (agent-identifier
1092       :name PSA@paracom.com)
1093   :receiver (set
1094     (agent-identifier
1095       :name CAiM@paracom.com))
1096   :ontology fipa-nas
1097   :language fipa-sl
1098   :protocol fipa-request
1099   :content "(
1100     (action
1101       (agent-identifier
1102         :name CAiM@paracom.com)
1103       (activate (sequence
1104         (transport-protocol
1105           :name fipa.mts.mtp.wap.std
1106           :gw-addr wap://gateway.com:1234/acc))))))")
1108
```

1109 3. **Message 4 propose:** CAiM sends a propose message to the CAiG.

```
1110 (propose
1111   :sender
1112     (agent-identifier
1113       :name CAiM@paracom.com)
1114   :receiver (set
1115     (agent-identifier
1116       :name CAiG@gateway.com))
1117   :ontology fipa-nas
1118   :language fipa-sl
1119   :protocol fipa-propose
1120   :content "(
1121     (action
1122       (agent-identifier
1123         :name CAiM@paracom.com)
1124       (use
1125         (transports
1126           :send (sequence
1127             (transport-protocol
1128               :name fipa.mts.mtp.wap.std))
1129           :recv (sequence
1130             (transport-protocol
1131               :name fipa.mts.mtp.wap.std))))))
1132     true)")
1133
```

1134  
 1135 4. **Message 5 request, message 6 agree and message 7 inform:** CAiG requests the local AP description to find  
 1136 out if the `fipa.mts.mtp.wap.std` MTP is supported (see [FIPA00023]).

1137  
 1138 5. **Message (8) accept-proposal:** CAiG accepts CAiM's proposal to use the `fipa.mts.mtp.wap.std` MTP.

```
1139 (accept-proposal
1140   :sender
1141     (agent-identifier
1142       :name CAiG@gateway.com)
1143   :receiver (set
1144     (agent-identifier
1145       :name CAiM@paracom.com))
1146   :ontology fipa-nas
1147   :language fipa-sl
1148
```

```

l149 :protocol fipa-propose
l150 :content "(
l151   (action
l152     (agent-identifier :name CAiM@paracom.com)
l153     (use
l154       (transports
l155         :send (sequence (transport-protocol :name fipa.mts.mtp.wap.std))
l156         :recv (sequence (transport-protocol :name fipa.mts.mtp.wap.std))))))
l157   (transport-selection
l158     (transports
l159       :send (sequence (transport-protocol :name fipa.mts.mtp.wap.std))
l160       :recv (sequence (transport-protocol :name fipa.mts.mtp.wap.std))))))")
l161

```

6. Messages 9 and 9' setup and messages 10 and 10' setup-done: CAiM (CAiG) commands ACCiM (ACCiG) to setup the fipa.mts.mtp.wap.std MTP. As this is intra-platform communication between CAiM (CAiG) and ACCiM (ACCiG), this is an implementation issue.

7. Message 11 inform: CAiM returns the result to the PSA.

```

l167
l168 (inform
l169   :sender
l170     (agent-identifier
l171       :name CAiM@paracom.com)
l172   :receiver (set
l173     (agent-identifier
l174       :name PSA@paracom.com))
l175   :ontology fipa-nas
l176   :language fipa-sl
l177   :protocol fipa-request
l178   :content "(
l179     (result
l180       (action
l181         (agent-identifier :name CAiM@paracom.com)
l182         (activate
l183           (sequence
l184             (transport-protocol
l185               :name fipa.mts.mtp.wap.std
l186               :gw-addr wap://gateway.com:1234/acc))))
l187         (transport-protocol
l188           :name fipa.mts.mtp.wap.std :gw-addr wap://gateway.com:1234/acc))))")
l189

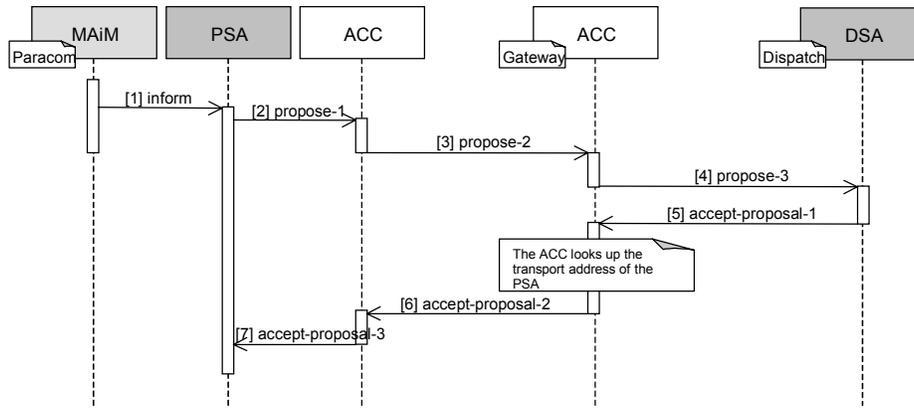
```

8. Message 12 and 12' set-description: CAiM (CAiG) modifies the AP description to show that the fipa.mts.mtp.wap.std is now active.

### l193 6.2.3 Example Negotiation of a Message Representation

l194 MAiM informs the PSA that the quality of the message transport connection has dropped significantly. The PSA reasons  
 l195 that the ACL representation needs to be changed to fipa.acl.rep.bitefficient.std and it proposes that to the  
 l196 DSA. The DSA accepts the PSA's proposal. The message flow is illustrated in *Figure 11*.

l197



**Figure 11: Example Negotiation of a Message Representation**

1. Message 1 `inform`: The MA informs the PSA that the outbound throughput has changed.

```

l198
l199
l200
l201
l202
l203
l204 (inform
l205   :sender
l206     (agent-identifier
l207       :name MAiM@paracom.com)
l208   :receiver (set
l209     (agent-identifier
l210       :name PSA@paracom.com))
l211   :ontology fipa-nas
l212   :language fipa-sl
l213   :protocol fipa-subscribe
l214   :conversation-id subscription-3106
l215   :content "(
l216     (qos-information
l217       (comm-channel name gprs)
l218       (qos :throughput
l219         (rate-value :unit Kbits/s :direction Outbound :value 0.96))))"
l220

```

2. Message 2 `propose-1`: Based on the new throughput value, the PSA decides to change to the message representation.

```

l221
l222
l223
l224 (propose
l225   :sender
l226     (agent-identifier
l227       :name PSA@paracom.com)
l228   :receiver (set
l229     (agent-identifier
l230       :name DSA@dispatch.com))
l231   :ontology fipa-nas
l232   :language fipa-sl
l233   :protocol fipa-propose
l234   :content "(
l235     (action
l236       (agent-identifier
l237         :name PSA@paracom.com)
l238       (use
l239         (msg-encoding
l240           :send (sequence
l241             (msg-representation
l242               :name fipa.acl.rep.bitefficient.std))
l243           :recv (sequence
l244             (msg-representation
l245               :name fipa.acl.rep.bitefficient.std))))))
l246     true)"

```

- 1247  
1248 3. Message 3 propose-2: The ACC at the mobile host forwards the same message to the ACC at the gateway host.  
1249  
1250 4. Message 4 propose-3: The ACC at the gateway host forwards the same message to the PSA.  
1251  
1252 5. Message 5 accept-proposal-1: The PSA accepts the proposal and sends a message back to the DSA.  
1253  
1254 (accept-proposal  
1255 :sender  
1256 (agent-identifier  
1257 :name DSA@dispatch.com)  
1258 :receiver (set  
1259 (agent-identifier  
1260 :name PSA@paracom.com))  
1261 :ontology fipa-nas  
1262 :language fipa-sl  
1263 :protocol fipa-propose  
1264 :content "(  
1265 (action  
1266 (agent-identifier :name PSA@paracom.com)  
1267 (use  
1268 (msg-encoding  
1269 :send (sequence  
1270 (msg-representation :name fipa.acl.rep.bitefficient.std))  
1271 :recv (sequence  
1272 (msg-representation :name fipa.acl.rep.bitefficient.std))))))  
1273 (msg-encoding-selection  
1274 (msg-encoding  
1275 :send (sequence  
1276 (msg-representation :name fipa.acl.rep.bitefficient.std))  
1277 :recv (sequence  
1278 (msg-representation :name fipa.acl.rep.bitefficient.std))))))")  
1279  
1280 6. Message 6 accept-proposal-2: The ACC at the gateway host forwards same message to the ACC at the  
1281 mobile host.  
1282  
1283 7. Message 7 accept-proposal-3: The ACC at the mobile host delivers the same message to the PSA.  
1284

## 7 References

- 1285
- 1286 [FIPA00023] FIPA Agent Management Specification. Foundation for Intelligent Physical Agents, 2000.  
1287 <http://www.fipa.org/specs/fipa00023/>
- 1288 [FIPA00036] FIPA Propose Interaction Protocol Specification. Foundation for Intelligent Physical Agents, 2000.  
1289 <http://www.fipa.org/specs/fipa00036/>
- 1290 [FIPA00069] FIPA ACL Message Representation in Bit-Efficient Encoding Specification. Foundation for Intelligent  
1291 Physical Agents, 2000.  
1292 <http://www.fipa.org/specs/fipa00069/>
- 1293 [FIPA00075] FIPA Agent Message Transport Protocol for IOP Specification. Foundation for Intelligent Physical  
1294 Agents, 2000.  
1295 <http://www.fipa.org/specs/fipa00075/>
- 1296 [FIPA00076] FIPA Agent Message Transport Protocol for WAP Specification. Foundation for Intelligent Physical  
1297 Agents, 2000.
- 1298 [FIPA00094] FIPA Quality of Service Specification. Foundation for Intelligent Physical Agents, 2000.  
1299 <http://www.fipa.org/specs/fipa00094/>
- 1300 [ITUE800] Recommendation E.800 - Telephone Network and ISDN, Quality of Service, Network Management and  
1301 Traffic Engineering, Terms and Definitions Related to Quality of Service and Network Performance  
1302 Including Dependability. International Telecommunication Union, International Telecommunication  
1303 Union, 1995.
- 1304 [ITUX135] Recommendation X.135 - Speed of Service (delay and throughput), Performance Values for Public  
1305 Data Networks when Providing Packet-Switched Services. International Telegraph and Telephone  
1306 Consultative Committee, 1993.
- 1307 [WAP99] Wireless Application Protocol Specification Version 1.2. WAP Forum, 1999.  
1308 <http://www.wapforum.org/what/technical.htm>  
1309

## 1310 8 Informative Annex A — ChangeLog

### 1311 8.1 2001/10/17 - version E by TC Gateways

1312	Page 8, lines 290-291:	Added a new frame <code>subscription-identifier</code> which is used to map
1313		subscriptions and subsequent cancel by the <code>subscribe-notification</code> and
1314		<code>cancel-notification</code> functions
1315	Page 12, lines 340-341:	Replaced predicate <code>qos-notification</code> with function <code>subscribe-notification</code> ;
1316		the <code>qos-notification</code> predicate was used as content for <code>subscribe act</code> , which
1317		is not used in this specification anymore, thus there is no need for this predicate, and,
1318		the <code>subscribe-notification</code> function replaces the <code>subscribe act</code> (in this spec),
1319		that is, it is used to subscribe changes in QoS
1320	Page 12, lines 341-342:	Added new function <code>cancel-notification</code> which replaces the <code>cancel act</code> (in this
1321		spec), that is, it is used to cancel previously subscribed notification(s)
1322	Page 13, lines 346-347:	Added sentence describing the return value of the function
1323	Page 14, lines 364-365:	Added a new refuse reason which is needed by the <code>cancel-notification</code> function
1324	Page 15, line 398:	Removed <code>fipa-subscribe</code> protocol from advertised protocols
1325	Pages 22-27, lines 799-1014:	“Message Exchange over WAP MTP” section removed because: (1) the example uses
1326		dynamic registration, and, (2) the functionality can be better implemented using FIPA
1327		messaging interoperability specification and FIPA message buffering specification
1328	Page 30, lines 1117-1119:	Figure 9 updated
1329	Page 30, lines 1127-1145:	Example ACL message updated to follow new subscription method
1330	Page 32, line 1216-1234:	Example ACL message updated to follow new subscription method
1331	Page 32, lines 1236-1268:	The <code>cancel</code> method replaced with the new one which includes replacing the <code>cancel</code>
1332		ACL message with <code>request</code> , <code>agree</code> and <code>inform</code> messages ( <code>fipa-request</code> )
1333	Page 34: lines 1268-1290:	The <code>subscribe</code> method replaced with the new one which includes replacing the
1334		<code>subscribe</code> ACL message with <code>request</code> , <code>agree</code> and <code>inform</code> messages ( <code>fipa-</code>
1335		<code>request</code> )
1336	Page 34, line 1290:	Updated message number
1337	Page 34, line 1310:	Updated message number
1338	Pages 34-35, lines 1312-1332:	Example ACL message updated to follow new subscription method
1339	Page 35, line 1334:	Updated message numbers
1340	Page 35, lines 1350-1368:	Example ACL message updated to follow new subscription method
1341	Page 38, lines 1496-1534:	Example ACL message updated to follow new subscription method
1342	Page 41, lines 1599-1600:	Removed reference to <code>fipa-subscribe</code> [FIPA00035]
1343		

### 1344 8.2 2002/09/13 - version F by TC X2S

1345	Entire document:	Changed all ontology terms to lowercase
1346	Entire document:	Ontology name changed from <code>FIPA-Nomadic-Application</code> to <code>fipa-nas</code>
1347	Entire document:	Examples updated according to other modifications
1348	Page 1, lines 102–103:	Removed reference to QoS ontology from the list of specification contents
1349	Page 1, lines 105–107:	Removed reference to WAP MTP and added references to bit-efficient message
1350		envelope and to QoS ontology specifications
1351	Page 2, lines 133–139:	Removed paragraph about WAP MTP
1352	Page 2, lines 160–161:	Removed reference to QoS ontology
1353	<b>Page 5, lines 266–268:</b>	<b>Removed the <code>qos</code> frame (moved to [FIPA00094])</b>
1354	<b>Page 6, lines 269–272:</b>	<b>Removed the <code>rate-value</code> frame (moved to [FIPA00094])</b>
1355	<b>Page 7, lines 273–276:</b>	<b>Removed the <code>time-value</code> frame (moved to [FIPA00094])</b>
1356	<b>Page 7, lines 277–280:</b>	<b>Removed the <code>probability-value</code> frame (moved to [FIPA00094])</b>
1357	<b>Page 8, lines 281–284:</b>	<b>Removed the <code>change-constraint</code> frame (moved to [FIPA00094])</b>
1358	<b>Page 8, lines 285–288:</b>	<b>Removed the <code>time-constraint</code> frame (moved to [FIPA00094])</b>
1359	<b>Page 8, lines 289–292:</b>	<b>Removed the <code>subscription-id</code> frame (moved to [FIPA00094])</b>
1360	<b>Page 8, lines 293–297:</b>	<b>Removed the <code>comm-channel</code> frame (moved to [FIPA00094])</b>

- 1361 **Page 9, lines 297–300:** **Removed the transport-protocol frame (moved to [FIPA00094])**
- 1362 **Page 11, lines 340–341:** **Removed the qos-information predicate (moved to [FIPA00094])**
- 1363 **Page 11, line 340:** **Added a transport-selection predicate**
- 1364 **Page 11, line 340:** **Added an msg-encoding-selection predicate**
- 1365 **Page 12, lines 343–344:** **Removed the subscribe-notification function (moved to [FIPA00094])**
- 1366 **Page 13, lines 345–346:** **Removed the cancel-notification function (moved to [FIPA00094])**
- 1367 Page 14, lines 362–364: Replaced the reference to the `fipa-agent-management not-understood`
- 1368 exception predicates with actual predicates
- 1369 Page 15, lines 366–368: Replaced the reference to the `fipa-agent-management refusal` exception
- 1370 propositions with the actual propositions
- 1371

### 1372 **8.3 2002/11/01 - version G by TC X2S**

- 1373 Entire document: Updated subscription examples to use `fipa-subscribe` protocol
- 1374

### 1375 **8.4 2002/12/03 - version H by FIPA Architecture Board**

- 1376 Entire document: Promoted to Standard status
- 1377