FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

FIPA Agent Message Transport Service Specification

Document title	FIPA Agent Message Transport Service Specification		
Document number	XC00067D	**Document source**	FIPA Agent Management
Document status	Experimental	**Date of this status**	2001/08/10
Supersedes	FIPA00024		
Contact	fab@fipa.org		
Change history			
2000/07/21	Approved for Experimental		
2001/08/10	Line numbering added		

Geneva, Switzerland

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.

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 Procedures for Technical Work. A complete overview of the FIPA specifications and their current status may be found in the FIPA List of Specifications. A list of terms and abbreviations used in the FIPA specifications may be found in the FIPA Glossary.

FIPA is a non-profit association registered in Geneva, Switzerland. As of January 2000, the 56 members of FIPA represented 17 countries worldwide. Further information about FIPA as an organization, membership information, FIPA specifications and upcoming meetings may be found at http://www.fipa.org/.

Contents

1 Scope

This document is part of the FIPA specifications and deals with message transportation between inter-operating agents. This document also forms part of the FIPA Agent Management specification (see [FIPA00023]) and contains specifications for agent message transport, including:

A reference model for an agent Message Transport Service, and,

Definitions for the expression of message transport information to an agent Message Transport Service.

2 Agent Message Transport Reference Model

2.1 Reference Model

The reference model for agent message transport comprises three levels (see *Figure 1*):

1. The Message Transport Protocol (MTP) is used to carry out the physical transfer of messages between two ACCs.

2. The Message Transport Service (MTS) is a service provided by the AP to which an agent is attached. The MTS supports the transportation of FIPA ACL messages between agents on any given AP and between agents on different APs.

3. The ACL represents the content of the messages carried by both the MTS and MTP.

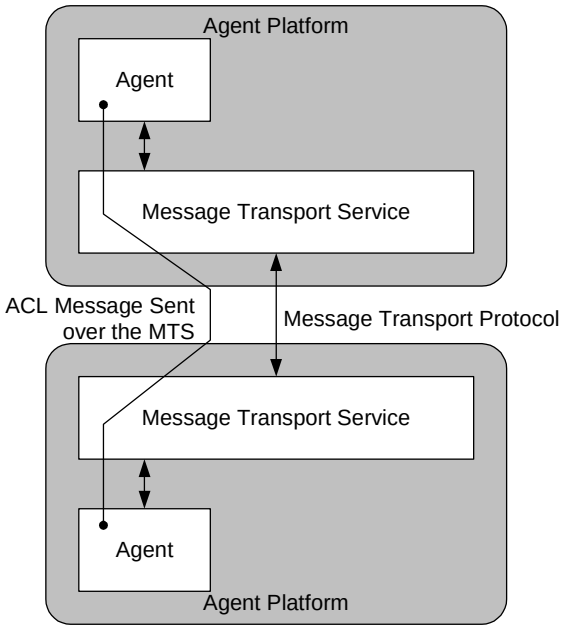

Figure 1: Message Transport Reference Model

2.2 Message Structure

In its abstract form, a message is made up of two parts: a message envelope expressing transport information and a message body comprising the ACL message of the agent communication.

For the purposes of message interpretation by an agent:

ACL semantics are defined only over the ACL message delivered in the message body of a FIPA message (see [FIPA00023]).

All information in the message envelope is supporting information only. How and if this information is used to by an agent for any kind of additional inference is undefined by FIPA.

3 Message Transport Service

The MTS provides a mechanism for the transfer of ACL messages between agents. The agents involved may be local to a single AP or on different APs. On any given AP, the MTS is provided by an Agent Communication Channel (ACC).

3.1 Message Envelope

Any MTP may use a different internal representation to describe a message envelope, but must express the same terms, represent the same semantics and perform the corresponding actions.

The following are general statements about the form of a message envelope:

A message envelope comprises a collection of parameters.

A parameter is a name/value pair.

A message envelope contains at least the mandatory `:to`, `:from`, `:date` and `:acl-representation` parameters.

A message envelope can contain optional parameters.

Each ACC handling a message may add new information to the message envelope, but it may never overwrite existing information. ACCs can add new parameters to a message envelope which override existing parameters that have the same parameter name; the mechanism for disambiguating message envelope entries is specified by each concrete message envelope syntax.

3.1.1 Updating Message Envelope Information

To update a value in one of the envelope parameters, the ACC must add a new copy of the message envelope parameter (containing the new value) to the envelope.

Since this mechanism permits multiple occurrences of the same parameters in a message envelope (with different values), each concrete message envelope syntax must provide a general mechanism for identifying which copy of the parameter is current. For example, the concrete envelope syntax given in [FIPA00073] specifies that the first occurrence of a parameter overrides any subsequent occurrence.

3.1.2 Additional Message Envelope Parameters

Any concrete syntax definition for the message envelope must include a clear mechanism for adding and distinguishing new and user defined parameters added to the message envelope. For example, the concrete envelope syntax given in [FIPA00073] specifies that all new and user defined parameters must be prefixed by “X-“.

3.2 Agent Identifiers and Transport Addresses

Agent Identifiers (AIDs) and transport addresses are defined in [FIPA00023].

3.3 Agent Communication Channel

The ACC is an entity providing a service directly to the agents on an AP. The ACC may access information provided by the other AP services (such as the AMS and DF) to carry out its message transport tasks.

3.3.1 Standard Interfaces

The standard MTP interfaces of an ACC are used to provide message transport interoperability between FIPA-compliant APs. To be FIPA-compliant, an ACC must have at least one such interface which supports a FIPA MTP. Furthermore, the ACC must support the FIPA baseline MTP for its AP description and may also provide other standard MTP interfaces (see section *3.5.2, Minimal Transport Requirements for Interoperability*).

When messages are received over a message interface advertised as implementing one of the FIPA standard MTPs, these messages must be handled as specified in section *3.3.3, Message Handling Behaviour*.

3.3.2 Proprietary Interfaces

FIPA does not specify how agents communicate using proprietary interfaces with the MTS.

3.3.3 Message Handling Behaviour

To provide the MTS, an ACC must transfer the messages it receives in accordance with the transport instructions contained in the message envelope. An ACC is only required to read the message envelope; it is not required to parse the message body. In performing message transfer tasks, the ACC may be required to obtain information from the AMS or DF on its own AP. Some implementations of ACCs may provide some form of buffering capability to help agents manage their messages.

3.3.4 Message Envelope Interpretation

The message forwarding behaviour of an ACC is determined by the instructions for message delivery that are expressed in the message envelope (see *Table 1*).

Parameter	Description
`to`	If no `:intended-receiver` parameter is present, then the information in this parameter is used to generate `:intended-receiver` field for the messages the ACC subsequently forwards.
`from`	If required, the ACC returns error and confirmation messages to the agent specified in this parameter.
`comments`	None.
`acl-representation`	None. This information is intended for the final recipient of the message.
`payload-length`	The ACC may use this information to improve parsing efficiency.
`payload-encoding`	None. This information is intended for the final recipient of the message.
`date`	None. This information is intended for the final recipient of the message.
`encrypted`	None. This information is intended for the final recipient of the message.
`intended-receiver`	An ACC uses this parameter to determine where this instance of a message should be sent. If this parameter is not provided, then the first ACC to receive the message should generate an `:intended-receiver` parameter using the `:to` parameter.
`received`	A new `:received` parameter is added to the envelope by each ACC that the message passes through. Each ACC handling a message must add a completed received parameter.
`transport-behaviour`	If present, the handling ACC must deliver the message according to the transport requirements specified in this parameter. If these requirements cannot be met or understood, then the ACC raises an error (see section *3.3.11, Error and Confirmation Messages*).

Table 1: Agent Communication Channel Interpretation of Message Envelope

3.3.5 Forwarding Messages

The recipients of a message are specified in the `:to` parameter of a message envelope and take the form of AIDs. Depending upon the presence of `:intended-receiver` parameters, the ACC forwards the message in one of the following ways:

If an ACC receives a message envelope without an `:intended-receiver`, then it generates a new `:intended-receiver` parameter from the `:to` parameter (possibly containing multiple AIDs). It may also generate multiple copies of the message with different `:intended-receiver` parameters if multiple receivers are specified. The `:intended-receiver` parameters form a delivery path showing the route that a message has taken.

If an ACC receives a message envelope with an `:intended-receiver` parameter, this is used for delivery of this instance of the message and the `:to` parameter is ignored.

If an ACC receives a message envelope with more than one `:intended-receiver` parameter, the most recent is used.

Before forwarding the message, the ACC adds a completed `:received` parameter to the message envelope. Once an ACC has forwarded a message it no longer needs to keep any record of the existence of that message.

3.3.6 Handling a Single Receiver

In delivering a message to a single receiver specified in the `:to` or `:intended-Receiver` parameters of a message envelope, the ACC forwards the message to one of the addresses in the `:addresses` parameter of the AID. If this address leads to another ACC, then it is the task of the receiving ACC to deliver the message to the receiving agent (if the agent is resident on the local AP) or to forward it on to another ACC (if the agent is not locally resident).

3.3.7 Handling Multiple Transport Addresses for a Single Receiver

The AID given in the `:to` or `:intended-receiver` parameter (in the case of both parameters being present, the information in the `:intended-receiver` parameter is used) of an message envelope may contain multiple transport addresses for a single receiving agent. The ACC uses the following method to try to deliver the message:

Try to deliver the message to the *first* transport address in the `:addresses` parameter; the first is chosen to reflect the fact that the transport address list in an AID is ordered by preference.

If this fails, because the agent or AP was not available or because the ACC does not support the appropriate message transport protocol, etc., then the ACC creates a new `:intended-receiver` parameter containing the AID with the failed transport address removed. The ACC then attempts to send the message to the next transport address in AID in the intended receiver list (now the first in the newly created `:intended-receiver` parameter).

If delivery is still unsuccessful when all transport addresses have been tried (or the AID contained no transport addresses), the ACC may try to resolve the AID using the name resolution services listed in the `:resolvers` parameter of the AID. Again, the name resolution services should be tried in the order of their appearance.

Finally, if all previous message delivery attempts have failed, then an appropriate error message for the final failure is passed back to the sending agent (see section *3.3.11, Error and Confirmation Messages*).

3.3.8 Handling Multiple Receivers

An ACC uses the following rules in delivering messages to multiple intended receivers[1]:

[1] An ACC may decide to optimise the delivery of messages where a given message is intended for multiple receivers that reside on the same host. However, whether an ACC decides to make this optimisation or not, the semantics of message delivery within an ACC must remain the same. This is so that optimised ACCs and non-optimised ACCs can inter-operate.

If an ACC receives a message envelope with no `:intended-receiver` parameter and a `:to` parameter containing more than one AID, it may or may not split these up to form separate messages[2]. Each message would contain a subset of the agents named in the `:to` and `:intended-receiver` parameters.

If an ACC receives a message envelope with an `:intended-receiver` parameter containing more than one AID, it may or may not split these up to form separate messages.

The resulting messages are handled as in the single receiver case (see section *3.3.6, Handling a Single Receiver*).

3.3.9 Delivering Messages

Once a message has arrived at ACC which can directly deliver it to the agent or agents named in the `:intended-receiver` parameter of the message envelope, this ACC should pass the message to the agents concerned. This specification does not specify how final message delivery is performed; the message may be passed to the agents using any of the ACC proprietary or standard MTP interfaces. An ACC should deliver the whole message, including the message envelope, to the receiving agent. However, particular AP implementations may provide middleware layers to free agents from the task of processing the envelope.

3.3.10 Using a Name Resolution Services

In certain circumstances, if an AID for a receiver contains no transport addresses then the ACC may try to resolve the AID by contacting one of the entities listed in the `:resolvers` parameter of the AID. The interface used by the ACC to do this is not specified by FIPA.

3.3.11 Error and Confirmation Messages

Error and confirmation messages sent to *a sending agent* by the MTS are in the form of ACL messages over the MTS. These MTS information messages are sent on behalf of the AMS agent responsible (the `:sender` parameter of the message must be set the local AMS's AID) of the ACC's AP. How the message is generated (whether by the AMS or by the ACC on behalf of the AMS) is not specified by FIPA.

If an error message needs to be returned, the message generated must follow the exception model defined [FIPA00023] such that:

The communicative act is a `failure`,

The predicate symbol is `internal-error`, and,

The argument parameter is a string describing the error which occurred (the form and content of which is not defined here).

3.4 Using the Message Transport Service

3.4.1 Sending Messages

An agent has three options when sending a message to another agent resident on a remote AP (see *Figure 2*):

1. Agent A sends the message to its local ACC using a proprietary or standard interface. The ACC then takes care of sending the message to the correct remote ACC using a suitable MTP. The remote ACC which will eventually deliver the message.

[2] Not splitting up messages may be more efficient when several copies would be delivered to the same address.

2. Agent A sends the message directly to the ACC on the remote AP on which Agent B resides. This remote ACC then delivers the message to B. To use this method, Agent A must support access to one of the remote ACC's MTP interfaces.

3. Agent A sends the message directly to Agent B, by using a direct communication mechanism. The message transfer, addressing, buffering of messages and any error messages must be handled by the sending and receiving agents. This communication mode is not covered by FIPA.

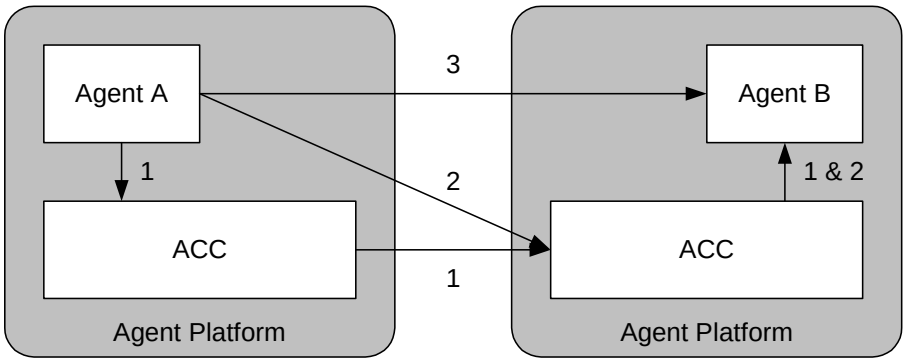

Figure 2: Three Methods of Communication between Agents on Different Agent Platforms[3]

3.4.2 Receiving Messages

An agent receives an entire message including both the message envelope and message body. Consequently, the receiving agent has access to all of the message transport information expressed in the message envelope, such as encryption details, ACL representation information, the delivery path of the message, etc.

3.5 Querying Message Transport Service Polices and Capabilities

An AP must support queries about its message transport policies and capabilities. Information pertinent to the MTS (such as the particular MTPs supported by an ACC) is given in the `:transport-profile` parameter of the AP description (see [FIPA00023]). An AP description can be accessed by sending a `get-description` request to an AMS.

3.5.1 Agent Platform Transport Descriptions

The transport description forms part of an AP and is expressed in `FIPA-SL0`. The following transport description is for an AP which supports IIOP and WAP based transport.:

```
(ap-transport-description
  :available-mtps
    (set
      (mtp-description
        :mtp-name fipa.mts.mtp.iiop.std
        :addresses (sequence iiop://foo.com/acc))
      (mtp-description
        :mtp-name fipa.mts.mtp.wap.std
        :addresses (sequence http://foo.com/acc http://bar.com/acc))))
```

For more information on how to generate a concrete representation of a transport description, see [FIPA00061] and [FIPA00008].

[3] A fourth possibility (not illustrated) is that instead of completing the last two stages of the first path, the ACC on the first platform contacts Agent B directly – this depends upon the address that the ACC is delivering to.

316 ### 3.5.2 Minimal Transport Requirements for Interoperability

317 To promote interoperability, FIPA mandates certain minimum transport capabilities for APs. The minimal transport
318 requirements for interoperability are classified by type of network environment an AP has access to and are grouped
319 into named interoperability transport profiles (see [FIPA00077] and [FIPA00078]). Each named transport profile defined
320 here has a name[4], a description and a single baseline MTP.
321
322

[4] Note that there is no ordering intended on the profiles defined in this section.

4 Agent Message Transport Ontology

4.1 Object Descriptions

This section describes a set of frames, that represent the classes of objects in the domain of discourse within the framework of the `FIPA-Agent-Management` ontology.

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

Frame. This is the mandatory name of this entity, that must be used to represent each instance of this class.

Ontology. This is the name of the ontology, whose domain of discourse includes the parameters described in the table.

Parameter. This is the mandatory name of a parameter of this frame.

Description. This is a natural language description of the semantics of each parameter.

Presence. This indicates whether each parameter is mandatory or optional.

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

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

4.1.1 Message Envelope Description

Frame	`envelope`
Ontology	`FIPA-Agent-Management`

Parameter	**Description**	**Presence**	**Type**	**Reserved Values**
`to`	This contains the names of the primary recipients of the message.	Mandatory	Sequence of `agent-identifier`	
`from`	This is the name of the agent who actually sent the message.	Mandatory	`agent-identifier`	
`comments`	This is a comment in the message envelope.	Optional	`String`	
`acl-representation`	This is the name of the syntax representation of the message body.	Mandatory	`String`	
`payload-length`	This contains the length of the message body.	Optional	`String`	

`payload-encoding`	This contains the language encoding of the message body	Optional[5]	`String`	`US-ASCII`
				`ISO-8859-1`
				…
				`ISO-8859-9`
				`UTF-8`
				`Shift_JIS`
				`EUC-JP`
				`ISO-2022-JP`
				`ISO-2022-JP-2`

`date`	This contains the creation date and time of the message envelope – added by the sending agent.	Mandatory	`Date`	

[5] If this field is not present, the default value `US-ASCII` is assumed for the content encoding.

encrypted	This contains information indicating how the message body has been encrypted.	Optional	Sequence of String	See [RFC822]
intended-receiver	This is the name of the agent to whom this instance of a message is to be delivered.	Optional	Sequence of agent-identifier	
received	This is a stamp representing the receipt of a message by an ACC.	Optional	received-object	
transport-behaviour	This contains the transport requirements of the message.	Optional	(Undefined)	

345

346 **4.1.2 Received Object Description**

Frame Ontology	received-object FIPA-Agent-Management			
Parameter	**Description**	**Presence**	**Type**	**Reserved Values**
by	The URL of the receiving ACC.	Mandatory	URL	
from	The URL of the sending ACC.	Optional	URL	
date	The date when a message was received.	Mandatory	Date	
id	The unique identifier of a message.	Optional	String	
via	The type of MTP the message was delivered over.	Optional	String	

347

348 **4.1.3 Agent Platform Transport Description**

Frame Ontology	ap-transport-description FIPA-Agent-Management			
Parameter	**Description**	**Presence**	**Type**	**Reserved Values**
available-mtps	A list of MTPs supported by the AP.	Optional	Set of mtp-description	

349

350 **4.1.4 Message Transport Protocol Description**

Frame Ontology	mtp-description FIPA-Agent-Management			
Parameter	**Description**	**Presence**	**Type**	**Reserved Values**
profile	The name of the FIPA transport profile.	Optional	String	See section *3.5.2*.
mtp-name	The FIPA name of the MTP being supported	Optional	String	
addresses	A list of the transport addresses of this MTP.	Mandatory	Sequence of URL	

351
352
353

5 References

[FIPA00007] FIPA Content Languages Specification. Foundation for Intelligent Physical Agents, 2000.
http://www.fipa.org/specs/fipa00007/

[FIPA00008] FIPA SL Content Language Specification. Foundation for Intelligent Physical Agents, 2000.
http://www.fipa.org/specs/fipa00008/

[FIPA00014] FIPA Nomadic Application Support Specification. Foundation for Intelligent Physical Agents, 2000.
http://www.fipa.org/specs/fipa00014/

[FIPA00023] FIPA Agent Management Specification. Foundation for Intelligent Physical Agents, 2000.
http://www.fipa.org/specs/fipa00023/

[FIPA00061] FIPA Agent Communication Language Specification. Foundation for Intelligent Physical Agents, 2000.
http://www.fipa.org/specs/fipa00061/

[FIPA00073] FIPA Agent Message Transport Envelope Representation in String Specification. Foundation for Intelligent Physical Agents, 2000.
http://www.fipa.org/specs/fipa00073/

[ISO8601] Date Elements and Interchange Formats, Information Interchange-Representation of Dates and Times. International Standards Organisation, 1998.
http://www.iso.ch/cate/d15903.html

[RFC822] Uniform Resource Identifiers: Generic Syntax. Request for Comments, 1992.
http://www.ietf.org/rfc/rfc0822.txt

[RFC2396] Standard for the Format of APRA Internet Text Messages. Request for Comments, 1998.
http://www.ietf.org/rfc/rfc2396.txt