FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

FIPA Messaging Interoperability Service Specification

Document title	FIPA Messaging Interoperability Service Specification		
Document number	XC00093A	**Document source**	FIPA TC Gateways
Document status	Experimental	**Date of this status**	2002/05/10
Supersedes	None		
Contact	fab@fipa.org		
Change history			
2002/05/10	Approved for Experimental		

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.

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 conversion between inter-operating agents. This document also forms part of the FIPA Message Transport Service Specification [FIPA00067] and contains specification for:

- FIPA Message conversion between different Message Transport Protocols or/and concrete encoding.

The document provides a series of examples to illustrate the agent management functions defined.

2 Overview

The FIPA Messaging Interoperability Service (FIPA-MIS) provides a means for converting between Message Transport Protocols (MTPs) and between concrete encodings of FIPA-message parts. FIPA-MIS can be used where direct end-to-end interoperability is impossible, impractical or undesirable. Direct end-to-end interoperability is impossible when communicating platforms/agents do not support any common message transport protocol or encoding of FIPA-message components, for example. Direct end-to-end interoperability may be impractical when communicating over a slow wireless link with a peer in the fixed network that does not support any message transport protocol suitable for wireless links.

2.1 Reference Model

The reference model for FMIS comprises four levels (see *Figure 1*):

1. The Message Transport Protocol Gateway (MTP-GW) is used to translate between Message Transport Protocols. For example, the Message Transport Protocol Gateway may translate between `fipa.mts.mtp.iiop.std` and `fipa.mts.mtp.wap.std`.

2. The Message Envelope Encoding Gateway (ENV-GW) is used to translate between Message Envelope encodings. For example, the Message Envelope Encoding Gateway may translate between `fipa.mts.env.rep.xml.std` and `fipa.mts.env.rep.bitefficient.std`.

3. The ACL Encoding Gateway (ACL-GW) is used to translate between ACL encodings. For example, the ACL Encoding Gateway may translate between `fipa.acl.rep.xml.std` and `fipa.acl.rep.bitefficient.std`.

4. The Content Language Encoding Gateway (CL-GW) is used to translate between Content Language encodings. Note that the current specification does not allow conversion between *different* content languages, only between *different encodings* of the same content language[1]. However, if this kind of functionality is needed, in can be added easily to the gateway specification. How such a translation is actually performed is an application implementation issue, and hence is out of scope.

The services specified here may also provide other kinds of translations (e.g., application dependent translation, etc.). This kind of functionality, however, should not be specified by FIPA, but hooks for such services exist in the specification.

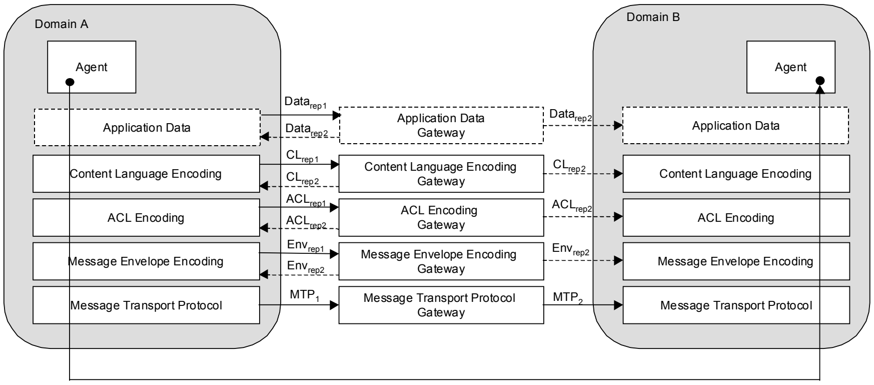

Figure 1: FIPA Messaging Interoperability Service Reference Model

[1] However, currently there is no content language specified in the FIPA Content Language Library that has more than one concrete encoding.

3 FIPA Messaging Interoperability Service

3.1 Requesting a Translation Service

When an ACC (or another gateway) finds out that some or all parts of a message or a MTP must be converted to another, it must first find a messaging interoperability service that can perform the necessary translations (this process is not defined here). After this, the functions provided by the service can be used in order to translate between message components (i.e., content language, ACL, or envelope). If translation of message transport protocol is needed, the message can be sent to the service that provides MTP-GW. The service knows implicitly the target MTP by examining the transport address of the destination agent. For example, let’s assume that the agent-identifier of the destination agent is as follows:

```
(agent-identifier
  :name foo@helluli.com
  :addresses (sequence (wap://helluli.com http://helluli.com/acc)))
```

When receiving the message using the message transport protocol, for example IIOP, the MTP-GW translates the message transport protocol to WAP.

3.1.1 Receiver Initiated Translation Service

When an agent knows in advance that it is not able to receive messages encoded in a particular encoding, it may request the messaging interoperability service to automatically translate all the messages directed to it. The agent sends a description of the encoding it is able to understand to the FIPA-MIS, which will translate the message with the suggested encoding.

4 Messaging Interoperability Service 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-MIS` 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 Translation Identifier

This type of object represents the unique identification for the incoming message translation.

Frame	`translation-id`
Ontology	`FIPA-MIS`

Parameter	**Description**	**Presence**	**Type**	**Reserved Values**
`Id`	Unique identifier for the incoming message translation. The identifier is unique only in one Messaging Interoperability Service.	Mandatory	`String`	

4.2 Function Descriptions

The following tables define usage and semantics of the functions that are part of the `FIPA-MIS` ontology.

The following terms are used to describe the functions of the `FIPA-MIS` domain:

Function. This is the symbol that identifies the function in the ontology.

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

Supported by. This is the type of agent that supports this function.

Description. This is a natural language description of the semantics of the function.

Domain. This indicates the domain over which the function is defined. The arguments passed to the function must belong to the set identified by the domain.

Range. This indicates the range to which the function maps the symbols of the domain. The result of the function is a symbol belonging to the set identified by the range.

Arity. This indicates the number of arguments that a function takes. If a function can take an arbitrary number of arguments, then its arity is undefined.

4.2.1 Available Encodings

Function	`available-encodings`
Ontology	`FIPA-MIS`
Supported by	`fipa-mis`
Description	An agent may query the service to provide a list of all encoding representations known by the service.
Domain	None
Range	`gateway-description`
Arity	`0`

4.2.2 Resolve Encoding

Function	`Resolve`
Ontology	`FIPA-MIS`
Supported by	`fipa-mis`
Description	An agent may query the service to resolve the encoding with which the supplied `message-component` has been encoded. If the action is successful, the service will return the `encoding-representation` of supplied message-component.
Domain	`message-component`[2]
Range	`encoding-representation`
Arity	`1`

4.2.3 Transform Encoding

Function	`transform`
Ontology	`FIPA-MIS`
Supported by	`fipa-mis`
Description	An agent may request the service to convert a transport-message or message component (i.e., payload, message, or content) into a particular encoding representation. The source message component is given as a parameter `message-component` and the `encoding-representation` parameter defines the target encoding. If the action is successful, the service will return the encoded message component.
Domain	`message-component`[2], `encoding-representation`
Range	`message-component`[2]
Arity	`2`

[2] The concrete syntax of the `message-component` depends on the concrete representation of the message component.

4.2.4 Request Incoming Translation

Function	`incoming-translation`
Ontology	`FIPA-MIS`
Supported by	`fipa-mis`
Description	An agent may request the service to convert automatically a transport-message or a message component (i.e., payload, message, or content) of an incoming message into a particular encoding representation before having it delivered. The preferred encoding is described in the `gateway-behaviour.` If the action is successful the service will return a `translation-id`, which can be used to cancel the translation service.
Domain	Sequence of `gateway-behaviour` (see [FIPA00067])
Range	`translation-id`
Arity	1

4.2.5 Cancel Incoming Translation

Function	`cancel-incoming-translation`
Ontology	`FIPA-MIS`
Supported by	`fipa-mis`
Description	An agent may request the service to stop transforming messages before delivering them to the agent.
Domain	`translation-id`
Range	The execution of this function results in a change of the state, but it has no explicit result. Therefore there is no range set.
Arity	1

4.3 Exceptions

The exceptions for the `FIPA-MIS` ontology follow the same form and rules as specified in [FIPA00023].

4.3.1 Not Understood Exception Propositions

The same set of *"Not Understood Exception Proposition*s" as in the `FIPA-Agent-Management` ontology is used in the `FIPA-MIS` ontology (see [FIPA00023]).

4.3.2 Refusal Exception Propositions

The same set of *"Refusal Exception Proposition*s" as defined in the `FIPA-Agent-Management` ontology is used in `FIPA-MIS` ontology (see [FIPA00023]). In addition, the `FIPA-MIS` ontology defines the propositions given below.

Communicative Act **Ontology**	`refuse` `FIPA-MIS`	
Predicate symbol	**Arguments**	**Description**
`Invalid-message`		The message component to be encoded is invalid in some way.
`Invalid-encoding`		The encoding-representation selected is unavailable.
`Unidentifiable-encoding`		The encoding-representation is unidentifiable by the service

4.3.3 Failure Exception Propositions

Communicative Act **Ontology**	failure FIPA-MIS

Predicate symbol	**Arguments**	**Description**
internal-error	String	See [FIPA00023].
unknown-identifier	String	The translation-id is unknown.

5 Registration of a FIPA Messaging Interoperability Service with the DF

In order for a FIPA messaging interoperability service to advertise its willingness to provide its services to an agent domain, it must register with a DF (as described in [FIPA00023]).

As part of this registration process the following constant values are introduced that universally identify the services the agent provides:

> The `type` slot in the `service-description` frame of FIPA messaging interoperability service must be declared as a constant `fipa-mis`.
>
> The `ontology` slot in the `service-description` frame of FIPA messaging interoperability service must be declared as a constant `FIPA-MIS`.

Below is given an example content of an agent `df-agent-description` frame which provides the following functionality:

> Translation service from XML encoded envelopes to bit-efficient envelopes, and,
>
> Translation service from XML encoded ACL messages to bit-efficient ACL messages.

```
(df-agent-description
  :name
    (agent-identifier
      :name fipa-gateway@iiop://foo.com/acc
      :addresses (sequence iiop://foo.com/acc))
  :ontology (set FIPA-MIS)
  :language (set FIPA-SL0)
  :services (set
    (service-description
      :name fipa-messaging-interoperability-service
      :type fipa-mis
      :ontology FIPA-MIS
      :properties
        (gateway-description
          :acl-translation
            (acl-gateway-description
              :from
                (encoding-representation :name fipa.acl.rep.xml.std)
              :to
                (set
                  (encoding-representation :name fipa.acl.rep.bitefficient.std)))
          :envelope-translation
            (envelope-gateway-description
              :from
                (encoding-representation :name fipa.mts.env.rep.xml.std)
              :to
                (set
                  (encoding-representation
                    :name fipa.mts.env.rep.bitefficient.std)))))
           :ownership (set Helluli)))))
```

6 References

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

[FIPA00067] FIPA Agent Message Transport Service Specification. Foundation for Intelligent Physical Agents, 2000. http://www.fipa.org/specs/fipa00067/

7 Informative Annex A — Examples

7.1 Transformation Encoding Request

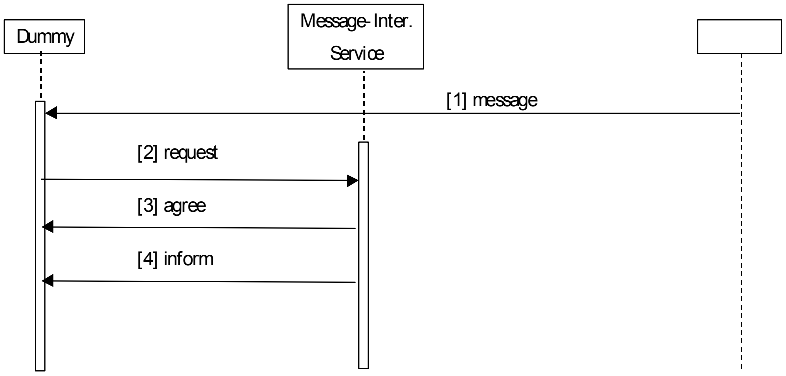

Figure 2: Transformation of message-component encoding

This example shows how an agent requests the Messaging Interoperability Service to transform a message component from one encoding to another. The message flow is illustrated in *Figure 2*.

1. Message [1]: The agent *dummy* receives a message and wants to transform the ACL-encoding of the message.

2. Message [2] `request`: The agent *dummy* sends the transform request to the Messaging Interoperability Service. The request contains the message-component to be transformed and the requested new encoding representation.

```
(request
  :sender
    (agent-identifier
      :name dummy
      :addresses (sequence http://helluli.com/acc))
  :receiver (set
    (agent-identifier
      :name fipa-messaging-interoperability-service
      :addresses (sequence http://fmis.com/acc)))
  :ontology FIPA-MIS
  :language FIPA-SL0
  :protocol fipa-request
  :content
    (action
      (agent-identifier
        :name fipa-messaging-interoperability-service)
      (transform
        (message-component (request …) )
        (encoding-representation
          :name fipa.acl.rep.bitefficient.std))))
```

3. Message [3] `agree`: The Messaging Interoperability Service agrees to perform the transformation:

```
(agree
  :sender
    (agent-identifier
      :name fipa-messaging-interoperability-service
      :addresses (sequence http://fmis.com/acc))
  :receiver (set
    (agent-identifier
      :name dummy
      :addresses (sequence http://helluli.com/acc)))
  :ontology FIPA-MIS
  :language FIPA-SL0
  :protocol fipa-request
  :content
    ((action
      (agent-identifier
         :name fipa-messaging-interoperability-service)
      (transform
        (message-component (request …) )
     (encoding-representation
          :name fipa.acl.rep.bitefficient.std)))
       true))
```

4. Message [4] `inform`: The Messaging Interoperability Service returns the encoded message component to the agent.

```
(inform
  :sender
    (agent-identifier
      :name fipa-messaging-interoperability-service
      :addresses (sequence http://fmis.com/acc))
  :receiver (set
    (agent-identifier
      :name dummy
      :addresses (sequence http://helluli.com/acc)))
  :ontology FIPA-MIS
  :language FIPA-SL0
  :protocol fipa-request
  :content
    (result
      (action
        (agent-identifier
           :name fipa-messaging-interoperability-service)
        (transform
          (message-component (request …) )
        (encoding-representation
            :name fipa.acl.rep.bitefficient.std)))
     (message-component 0xfa… )))
```

7.2 Resolve Encoding

This example shows how an agent requests the Messaging Interoperability Service to resolve the encoding of a message component.

1. Message [1] `request`: The agent *dummy* sends the resolve request to the Messaging Interoperability Service:

```
(request
  :sender
    (agent-identifier
      :name dummy
      :addresses (sequence http://helluli.com/acc))
  :receiver (set
    (agent-identifier
      :name fipa-messaging-interoperability-service
      :addresses (sequence http://fmis.com/acc)))
  :ontology FIPA-MIS
  :language FIPA-SL0
  :protocol fipa-request
  :content
    (action (agent-identifier :name fipa-messaging-interoperability-service)
      (resolve
        (message-component <fipa-message>…</fipa-message>))))
```

2. Message [2] `agree`: The Messaging Interoperability Service agrees to perform the action.

3. Message [3] `inform:` The Messaging Interoperability Service informs the agent *dummy* that the message is encoded using `fipa.acl.rep.xml.std`.

```
(inform
  :sender
   (agent-identifier
      :name fipa-messaging-interoperability-service
      :addresses (sequence http://fmis.com/acc))
  :receiver (set
    (agent-identifier
      :name dummy
      :addresses (sequence http://helluli.com/acc)))
  :ontology FIPA-MIS
  :language FIPA-SL0
  :protocol fipa-request
  :content
   (result
    (action (agent-identifier :name fipa-messaging-interoperability-service)
      (resolve
        (message-component <fipa-message>…</fipa-message>)))
      (encoding-representation
         :name fipa.acl.rep.xml.std)))
```

7.3 Receiver initialised transformations

This example shows how an agent requests the Messaging Interoperability Service to transform messages before their delivery to the agent.

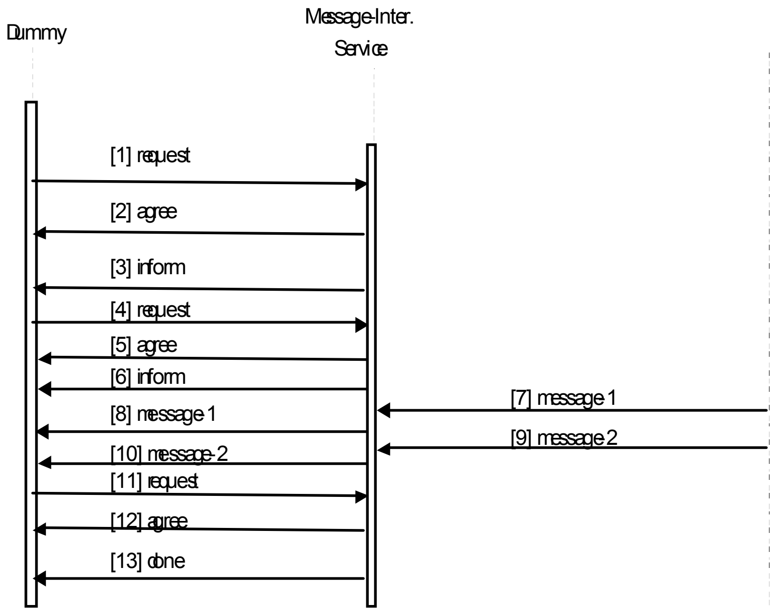

Figure 3: Receiver Initialised Transformations

1. Message [1] `request`: The agent *dummy* query the Messaging Interoperability Service a list of all the encoding representations known by the service.

```
(request
  :sender
    (agent-identifier
      :name dummy
      :addresses (sequence http://campa.com/acc))
  :receiver (set
    (agent-identifier
      :name fipa-messaging-interoperability-service
      :addresses (sequence http://fmis.com/acc)))
  :ontology FIPA-MIS
  :language FIPA-SL0
  :protocol fipa-request
  :content
    (action
      (agent-identifier
        :name fipa-messaging-interoperability-service)
      (available-encodings)))
```

2. Message [2] `agree`: The Messaging Interoperability Service agrees to deliver the list.

3. Message [3] `inform`: The Messaging Interoperability Service sends the list:

```
(inform
  :sender
   (agent-identifier
      :name fipa-messaging-interoperability-service
      :addresses (sequence http://fmis.com/acc))
  :receiver (set
    (agent-identifier
      :name dummy
      :addresses (sequence http://campa.com/acc)))
  :ontology FIPA-MIS
  :language FIPA-SL0
  :protocol fipa-request
  :content
    (result
      (action
        (agent-identifier
          :name fipa-messaging-interoperability-service)
      (available-encodings))
    (gateway-description
   :acl-translation
     (set
          (acl-gw-description
            :from fipa.acl.rep.bitefficient.std
         :to (set fipa.acl.rep.string.std fipa.acl.rep.xml.std))
          (acl-gw-description
            :from fipa.acl.rep.string.std
         :to (set fipa.acl.rep.bitefficient.std))))))
```

4. Message [4] `request`: The agent *dummy* requests to the Messaging Interoperability Service to transform messages to the `fipa.acl.rep.bitefficient.std` encoding before delivering them to the agent *dummy*:

```
(request
  :sender
    (agent-identifier
      :name dummy
      :addresses (sequence http://campa.com/acc))
  :receiver (set
    (agent-identifier
      :name fipa-messaging-interoperability-service
      :addresses (sequence http://fmis.com/acc)))
  :ontology FIPA-MIS
  :language FIPA-SL0
  :protocol fipa-request
  :content
    (action (agent-identifier :name fipa-messaging-interoperability-service)
      (incoming-translation
     (sequence
          (gateway-behaviour
             :acl fipa.acl.rep.bitefficient.std)))))
```

5. Message [5] `agree`: The Messaging Interoperability Service agrees.

6. Message [6] `inform`: The Messaging Interoperability Service returns an translation identifier:

```
(inform
  :sender
   (agent-identifier
      :name fipa-messaging-interoperability-service
      :addresses (sequence http://fmis.com/acc))
  :receiver (set
    (agent-identifier
      :name dummy
      :addresses (sequence http://campa.com/acc)))
  :ontology FIPA-MIS
  :language FIPA-SL0
  :protocol fipa-request
  :content
    (result
     (action (agent-identifier :name fipa-messaging-interoperability-service)
      (incoming-translation
     (sequence
          (gateway-behaviour
             :acl fipa.acl.rep.bitefficient.std))))
     (translation-id :id id1)))
```

7. Message [7]: The service receives a message for *dummy*, and converts the ACL encoding to `fipa.acl.rep.bitefficient.std`.

8. Message [8]: The service delivers the message to the agent *dummy*.

9. Message [9] and Message [10]: Another message delivered to the agent *dummy* after being translated.

10. Message [11] `request`: The agent *dummy* sends a request to the Messaging Interoperability Service to cancel the translation of incoming messages:

```
(request
  :sender
    (agent-identifier
      :name dummy
      :addresses (sequence http://campa.com/acc))
  :receiver (set
    (agent-identifier
      :name fipa-messaging-interoperability-service
      :addresses (sequence http://fmis.com/acc)))
  :ontology FIPA-MIS
  :language FIPA-SL0
  :protocol fipa-request
  :content
    (action (agent-identifier :name fipa-messaging-interoperability-service)
      (received-translated-cancel
     (translation-id :id id1))))
```

11. Message [12] `agree:` The service agrees.

12. Message [13] `inform`: The service informs the agent that the translation of the incoming messages has been cancelled.