FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

FIPA Agent Message Transport Protocol for IIOP Specification

Document title	FIPA Agent Message Transport Protocol for IIOP Specification		
Document number	XC00075D	**Document source**	FIPA Agent Management
Document status	Experimental	**Date of this status**	2000/09/01
Supersedes	FIPA00024		
Contact	fab@fipa.org		
Change history			
2000/09/01	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 1
2 Message Transport Protocol for IIOP 2
2.1 Component Name 2
2.2 Interface Definition 2
2.3 ACC Processing of IDL Envelope 2
2.4 Concrete Message Envelope Syntax 3
3 References 6

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 [FIPA00023] and contains specifications for:

- The transportation of messages between agents using the Internet Inter-Orb Protocol (IIOP - see [OMGiiop]).

2 Message Transport Protocol for IIOP

This MTP is based on the transfer of an OMG IDL structure containing the message envelope and an octet sequence representing the ACL message body. The envelope and the message body are transferred together within a single IIOP one-way invocation [OMGiiop].

Once the request has been received, the message envelope is used by the ACC to obtain the instructions and information needed to correctly handle the message body.

2.1 Component Name

The name assigned to this component is:

```
fipa.mts.mtp.iiop.std
```

2.2 Interface Definition

The following IDL specifies the message transport interface. This interface contains a single operation message() that requires a single argument. This argument has two attributes: a sequence of Envelope structures holding the message envelope, and the payload, that is a sequence of octets containing the ACL message body.

```
module FIPA {
  typedef sequence<Envelope> Envelopes;
  typedef sequence<octet> Payload;
  struct FipaMessage {
    Envelopes messageEnvelopes;
    Payload   messageBody;
  };

  interface MTS {
    oneway void message(in FipaMessage aFipaMessage);
  };
};
```

2.3 ACC Processing of IDL Envelope

According to [FIPA00067], a FIPA compliant ACC is not allowed to modify any element of the envelope that it receives. It is however allowed to update a value in one of the envelope slots by adding a new Envelope element at the end of the `messageEnvelopes` sequence. This new element is required to have only those slot values that the ACC wishes to add or update plus a new `ReceivedObject` element as mandated in [FIPA00067].

As a consequence, an ACC that receives a message must implement the procedure described in the following pseudo-code. The procedure recomposes the full envelope structure with its latest values for each slot. The procedure simply shows that the ACC starts from the last envelope in the sequence and continues until it has all the required values for each slot of the envelope.

```
EnvelopeWithAllFields := new empty Envelope;

while ( (EnvelopeWithAllFields does not contain values for all its fields)
        OR (all Envelopes in the sequence have been processed) ) {
 // the ACC gets the next envelope in the sequence starting from the end
 tempEnvelope = getNextEnvelope;
 foreach field in an envelope {
   if ((this field has no value in envelopeWithAllFields)
       AND (this field has a value in tempEnvelope))
   then copy the value of this field from tempEnvelope to envelopeWithAllFields
 }
}
```

`EnvelopeWithAllFields` now contains the latest values for all its fields.

For example:

```
Envelope(0):
  to = tizio
  from = caio
  aclRepresentation = XML
  received = …

Envelope (1):
  from = caio@molfetta.it
  received = …

Envelope (2) :
  intended-receiver = tizio@villardora.it
  received = …

EnvelopeWithAllFields:
 to = tizio                                (from envelope 0)
 from = caio@molfetta.it                   (from envelope 1)
 intended-receiver = tizio@villardora.it   (from envelope 2)
 date = 25 May 2000                        (from envelope 0)
```

2.4 Concrete Message Envelope Syntax

The Abstract Envelope Syntax from [FIPA00067] maps into a set of OMG IDL structured types, all of which are enclosed within the FIPA module.

The following standard convention applies for the identification of optional slots: an empty string and an empty sequence identify the non-presence of a slot. In the case of payload-length, that is a number, any negative value can be used to identify the non-presence of the slot.

The complete IDL definition is:

```
module FIPA {
  // No need for an URL struct, since it's only put in the
  // message envelope for informational purposes.
  typedef string URL;

  typedef sequence<string> strings; // a sequence of strings

  // this generic type is used to represent user-defined, non FIPA-defined,
  // properties that are added to the message envelope in the form of a
  // keyword and value pair.
  struct Property {
    string keyword;
    any value;
  };

  struct AgentID {  // Agent Identifier
```

```
    string name;
    sequence<URL>      addresses;
    sequence<AgentID>  resolvers;
    sequence<Property> userDefinedProperties;
  };

  typedef sequence<AgentID> AgentIDs;  // sequence of Agent Identifiers

  // IDL struct to represent a time stamp.
  // It is based on the ISO8601 format with extension for millisecond durations.
  // The value of the typeDesignator must be a valid
  // AlphaCharacter, i.e. ['a'-'z' , 'A'-'Z'], that identifies the timezone.
  // ISO8601 reports the mapping between typeDesignator and timezone.
  // The typeDesignator for UTC is the character 'Z'.
  // If the value of typeDesignator is not an AlphaCharacter, it defaults
  // to the local timezone.
  struct DateTime {
    short year;  // year (e.g. 2000)
    short month; // between 1 and 12
    short day;   // between 1 and 31
    short hour;  // between 0 and 23
    short minutes; // between 0 and 59
    short seconds; // between 0 and 59
    short milliseconds; // between 0 and 999
    char  typeDesignator; // see comment above
  };

  struct ReceivedObject {
    URL by;
    URL from;
    DateTime date;
    string id;
    string via;
  };

  typedef  sequence<Property> TransportBehaviourType;

  typedef sequence<AgentID,1> OptAgentID;
  typedef sequence<DateTime,1> OptDateTime;
  typedef sequence<TransportBehaviourType,1> OptTransportBehaviourType;
  typedef sequence<ReceivedObject,1> OptReceivedObject;

  struct Envelope {
     AgentIDs                  to;
     OptAgentID                from;
     string                    comments;
     string                    aclRepresentation;
     long                      payloadLength;
     string                    payloadEncoding;
     OptDateTime               date;
     strings                   encrypted;
     AgentIDs                  intendedReceiver;
     OptReceivedObject         received;
     OptTransportBehaviourType transportBehaviour;
     sequence<Property>        userDefinedProperties; // user-defined properties
  };
  typedef sequence<Envelope> Envelopes;

  typedef sequence<octet> Payload;

  struct FipaMessage {
    Envelopes messageEnvelopes;
    Payload   messageBody;
  };

  interface MTS {
    oneway void message(in FipaMessage aFipaMessage);
  };
};
```

3 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/`

[OMGiiop] OMG Internet Inter-ORB Protocol Specification, Common Object Request Broker Architecture 2.2. Object Management Group, 1999.