

# FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

## FIPA Agent Message Transport Envelope Representation in XML Specification

<b>Document title</b>	FIPA Agent Message Transport Envelope Representation in XML Specification		
<b>Document number</b>	XC000851	<b>Document source</b>	FIPA Agent Management
<b>Document status</b>	Experimental	<b>Date of this status</b>	2002/10/18
<b>Supersedes</b>	None		
<b>Contact</b>	fab@fipa.org		
<b>Change history</b>	See <i>Informative Annex C — 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 organisation 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 organisations, which are companies  
24 and universities that are active in the field of agents. FIPA makes the results of its activities available to all interested  
25 parties 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 organisation without restriction. In particular, members are not bound  
29 to 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  
33 process of specification may be found in the FIPA Document Policy [f-out-00000] and the FIPA Specifications Policy [f-  
34 out-00003]. A complete overview of the FIPA specifications and their current status may be found on the FIPA Web  
35 site.

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

40 **Contents**

41	1	Scope .....	1
42	2	XML Envelope Representation.....	2
43	2.1	Component Name .....	2
44	2.2	Mime Type.....	2
45	2.3	Syntax .....	2
46	2.4	Additional Syntax Rules .....	3
47	2.5	Representation of Time.....	4
48	3	References .....	5
49	4	Informative Annex A — Examples.....	6
50	5	Informative Annex B — Notes .....	10
51	6	Informative Annex C — ChangeLog .....	11
52	6.1	2002/05/10 - version I by FIPA Architecture Board .....	11
53			

53 **1 Scope**

54 This document is part of the FIPA specifications and deals with message transportation between inter-operating  
55 agents. This document also forms part of the FIPA Agent Management Specification [FIPA00023] and contains  
56 specifications for:

- 57
- 58 • Syntactic representations of a message envelope in XML form (see [W3Cxml]).
- 59

60

## 60 2 XML Envelope Representation

61 This section gives the concrete syntax for the message envelope specification that must be used to transport  
 62 messages over a Message Transport Protocol (MTP - see [FIPA00067]). This concrete syntax is designed to  
 63 complement [FIPA00071] and [FIPA00084].  
 64

### 65 2.1 Component Name

66 The name assigned to this component is:

```
67
68 fipa.mts.env.rep.xml.std
69
```

### 70 2.2 Mime Type

71 Where required, the MIME type (see [RFC2046]) of items generated according to this specification is taken to be  
 72 application/xml. The `charset` encoding used in this section must conform to [W3Cxml].  
 73

### 74 2.3 Syntax

75 The following DTD specifies the encoding of the abstract FIPA specification as an XML message:

```
76
77 <!--
78 Document Type: XML DTD
79 Document Purpose: Encoding of FIPA ACL message envelopes (as in [FIPA0067]).
80 See http://www.fipa.org
81 Last Revised: 2000-08-16
82 -->
83
84 <!ELEMENT    envelope    ( params+ )>
85
86 <!ELEMENT    params      ( to?,
87                          from?,
88                          comments?,
89                          acl-representation?,
90                          payload-length?,
91                          payload-encoding?,
92                          date?,
93                          encrypted?,
94                          intended-receiver?,
95                          received?,
96                          user-defined* )>
97
98 <!ATTLIST    params      index CDATA #REQUIRED>
99
100 <!ELEMENT    to          ( agent-identifier+ )>
101
102 <!ELEMENT    from        ( agent-identifier )>
103
104 <!ELEMENT    acl-representation ( #PCDATA )>
105
106 <!ELEMENT    comments     ( #PCDATA )>
107
108 <!ELEMENT    payload-length ( #PCDATA )>
109
110 <!ELEMENT    payload-encoding ( #PCDATA )>
111
112 <!ELEMENT    date         ( #PCDATA )>
113
114 <!ELEMENT    intended-receiver ( agent-identifier+ )>
115
116
```

```

117 <!ELEMENT      agent-identifier      ( name,
118                                     addresses?,
119                                     resolvers?,
120                                     user-defined* )>
121
122 <!ELEMENT      name                   ( #PCDATA )>
123
124 <!ELEMENT      addresses              ( url+ )>
125
126 <!ELEMENT      url                    ( #PCDATA )>
127
128 <!ELEMENT      resolvers              ( agent-identifier+ )>
129
130 <!ELEMENT      received               ( received-by,
131                                     received-from?,
132                                     received-date,
133                                     received-id?,
134                                     received-via?,
135                                     user-defined* )>
136
137 <!ELEMENT      received-by            ( url )>
138
139 <!ELEMENT      received-from          ( url )>
140
141 <!ELEMENT      received-date          EMPTY>
142 <!ATTLIST      received-date         value CDATA #IMPLIED>
143
144 <!ELEMENT      received-id            EMPTY>
145 <!ATTLIST      received-id          value CDATA #IMPLIED>
146
147 <!ELEMENT      received-via          EMPTY>
148 <!ATTLIST      received-via        value CDATA #IMPLIED>
149
150 <!ELEMENT      user-defined           ( #PCDATA )>
151 <!ATTLIST      user-defined         href CDATA #IMPLIED>
152

```

## 2.4 Additional Syntax Rules

The following additional rules not specified in the DTD also apply:

1. [FIPA00067] requires that all changes made to a message envelope by one message processing step (for example, handling of the message by a single ACC) be attributable to the message processor that made the changes. This is achieved in the XML envelope by grouping all changes made by one message processor (ACC) at one point in time into a single PARAMS element.
2. There is no need to add envelope parameter values to a new PARAMS element if the values of these parameters is not being updated. Only parameters whose value is being changed need be included. The meaning of a PARAMS statement containing two elements defining new values for the same envelope parameter is undefined.
3. This specification permits multiple occurrences of unique message envelope-level parameters (to, from, intended-receiver, date, acl-representation, payload-length, received transport-behaviour, etc.) in order to handle field value overwriting as specified in [FIPA00067]. To help obtain the latest (and currently valid) value of any parameter, the INDEX attribute of the PARAMS element is used to establish an order of the different occurrences of elements (and hence envelope parameters). The first and oldest occurrence of the element will have an INDEX value of 1, the next value of the field will have INDEX value of 2 and so on.
4. When adding a new PARAMS element, the INDEX attribute will have a value with 1 higher than the largest existing INDEX of any PARAMS element currently in the envelope. The first PARAMS element will have the INDEX value of 1.

177 5. The current value of any envelope-level field will be given by the value of the field as it appears in the newest  
 178 PARAMS element that contains that field.

180 6. The following pseudo code algorithm may be used to obtain the latest values for each of the envelope parameters:

```

181 EnvelopeWithAllFields := new empty Envelope;
182
183 while ((EnvelopeWithAllFields does not contain values for all its fields)
184        OR (all PARAMS elements in the sequence have been processed))
185 {
186   // the processor gets the next envelope in the sequence starting with the one
187   // with the highest index
188   tempEnvelope = getNextEnvelope;
189
190   foreach field in an envelope
191   {
192     if ((this field has no value in envelopeWithAllFields)
193         AND (this field has a value in tempEnvelope))
194       then copy the value of this field from tempEnvelope to envelopeWithAllFields;
195   }
196 }
197
198 EnvelopeWithAllFields contains now the latest values for all its fields set in the envelope.
  
```

201 7. User-defined fields in the params, agent-identifier and received objects must be prefixed with 'x-'.  
 202

## 203 2.5 Representation of Time

204 Time tokens are based on [ISO8601], with extensions for relative time and millisecond duration's. Time expressions  
 205 may be absolute, or relative to the current time. If no type designator is given, the local time zone is used. The type  
 206 designator for UTC is the character Z. UTC is preferred to prevent time zone ambiguities. Note that years must be  
 207 encoded in four digits. As examples, 8:30am on April 15th, 1996 local time would be encoded as:

208  
 209 19960415T0830000000

210 The same time in UTC would be:

211  
 212 19960415T083000000Z  
 213  
 214  
 215

### 215 3 References

- 216 [FIPA00023] FIPA Agent Management Specification. Foundation for Intelligent Physical Agents, 2000.  
217 <http://www.fipa.org/specs/fipa00023/>
- 218 [FIPA00067] FIPA Agent Message Transport Service Specification. Foundation for Intelligent Physical Agents,  
219 2000. <http://www.fipa.org/specs/fipa00067/>
- 220 [FIPA00069] FIPA ACL Message Representation in Bit-Efficient Encoding Specification. Foundation for Intelligent  
221 Physical Agents, 2000.  
222 <http://www.fipa.org/specs/fipa00069/>
- 223 [FIPA00070] FIPA ACL Message Representation in String Specification. Foundation for Intelligent Physical Agents,  
224 2000.  
225 <http://www.fipa.org/specs/fipa00070/>
- 226 [FIPA00071] FIPA ACL Message Representation in XML Specification. Foundation for Intelligent Physical Agents,  
227 2000.  
228 <http://www.fipa.org/specs/fipa00071/>
- 229 [FIPA00075] Agent Message Transport Protocol for IOP. Foundation for Intelligent Physical Agents, 2000.  
230 <http://www.fipa.org/specs/fipa00075/>
- 231 [FIPA00084] FIPA Agent Message Transport Protocol for HTTP Specification. Foundation for Intelligent Physical  
232 Agents, 2000.  
233 <http://www.fipa.org/specs/fipa00084/>
- 234 [ISO8601] Date Elements and Interchange Formats, Information Interchange-Representation of Dates and  
235 Times. International Standards Organisation, 1998.  
236 <http://www.iso.ch/cate/d15903.html>
- 237 [RFC2046] Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, Freed and Borenstein,  
238 November 1996.  
239 <http://www.rfc-editor.org/rfc/rfc2046.txt>
- 240 [W3Cxml] Extensible Mark-up Language (XML) 1.0 Specification (Recommendation). World Wide Web  
241 Consortium, 1998.  
242 <http://www.w3c.org/TR/REC-xml/>  
243
- 244



## 4 Informative Annex A — Examples

1. Here is a simple example of an envelope conforming to the DTD described in section 2.3, *Syntax*:

```

244
245
246
247 <?xml version="1.0"?>
248 <envelope>
249   <params index="1">
250     <to>
251       <agent-identifier>
252         <name>receiver@foo.com</name>
253         <addresses>
254           <url>http://foo.com/acc</url>
255         </addresses>
256       </agent-identifier>
257     </to>
258     <from>
259       <agent-identifier>
260         <name>sender@bar.com</name>
261         <addresses>
262           <url>http://bar.com/acc</url>
263         </addresses>
264       </agent-identifier>
265     </from>
266
267     <acl-representation>fipa.acl.rep.xml.std</acl-representation>
268
269     <date>20000508T042651481</date>
270
271     <received >
272       <received-by value="http://foo.com/acc" />
273       <received-date value="20000508T042651481" />
274       <received-id value="123456789" />
275     </received>
276   </params>
277 </envelope>
278

```

2. Here is an example which covers all the aspects described in section 2.3, *Syntax*:

```

279
280
281 <?xml version="1.0"?>
282 <envelope>
283   <params index="1">
284     <to>
285       <agent-identifier>
286         <name>receiver@foo.com</name>
287         <addresses>
288           <url>http://foo.com/acc</url>
289         </addresses>
290       <resolvers>
291         <agent-identifier>
292           <name>resolver@bar.com</name>
293           <addresses>
294             <url>http://bar.com/acc1</url>
295             <url>http://://bar.com/acc2</url>
296             <url>http://bar.com/acc3</url>
297           </addresses>
298         </agent-identifier>
299       </resolvers>
300     </agent-identifier>
301   </to>
302
303   <from>
304     <agent-identifier>
305       <name>sender@bar.com</name>
306       <addresses>
307         <url>http://bar.com/acc</url>

```

```

308     </addresses>
309     <resolvers>
310       <agent-identifier>
311         <name>resolver@foobar.com</name>
312         <addresses>
313           <url>http://foobar.com/acc1</url>
314           <url>http://foobar.com/acc2</url>
315           <url>http://foobar.com/acc3</url>
316         </addresses>
317       </agent-identifier>
318     </resolvers>
319   </agent-identifier>
320 </from>
321
322 <comments>No comments!</comments>
323
324 <acl-representation>fipa.acl.rep.xml.std</acl-representation>
325
326 <payload-encoding>US-ASCII</payload-encoding>
327
328 <date>20000508T042651481</date>
329
330 <intended-receiver>
331   <agent-identifier>
332     <name>intendedreceiver@foobar.com</name>
333     <addresses>
334       <url>http://foobar.com/acc1</url>
335       <url>http://foobar.com/acc2</url>
336       <url>http://foobar.com/acc3</url>
337     </addresses>
338     <resolvers>
339       <agent-identifier>
340         <name>resolver@foobar.com</name>
341         <addresses>
342           <url>http://foobar.com/acc1</url>
343           <url>http://foobar.com/acc2</url>
344           <url>http://foobar.com/acc3</url>
345         </addresses>
346       <resolvers>
347         <agent-identifier>
348           <name>resolver@foobar.com</name>
349           <addresses>
350             <url>http://foobar.com/acc1</url>
351             <url>http://foobar.com/acc2</url>
352             <url>http://foobar.com/acc3</url>
353           </addresses>
354         </agent-identifier>
355       </resolvers>
356     </agent-identifier>
357   </resolvers>
358 </agent-identifier>
359 </intended-receiver>
360
361 <received>
362   <received-by value="http://foo.com/acc" />
363   <received-from value="http://foobar.com/acc" />
364   <received-date value="20000508T042651481" />
365   <received-id value="123456789" />
366   <received-via value="http://bar.com/acc" />
367 </received>
368 </params>
369 </envelope>
370

```

3. Here is an example which also includes the MIME multipart encapsulation which might be used over HTTP (see [FIPA00084]):

```

374 MIME-Version: 1.0

```

```

375 Content-Type: multipart-mixed ;
376     boundary="--251D738450A171593A1583EB"
377
378 This is not part of the MIME multipart encoded message.
379 --251D738450A171593A1583EB
380 Content-Type: application/xml
381
382 <?xml version="1.0"?>
383 <envelope>
384     <params index="1">
385         <to>
386             <agent-identifier>
387                 <name>receiver@foo.com</name>
388                 <addresses>
389                     <url>http://foo.com/acc</url>
390                 </addresses>
391             </agent-identifier>
392         </to>
393         <from>
394             <agent-identifier>
395                 <name>sender@bar.com</name>
396                 <addresses>
397                     <url>http://bar.com/acc</url>
398                 </addresses>
399             </agent-identifier>
400         </from>
401
402         <acl-representation>fipa.acl.rep.string.std</acl-representation>
403
404         <payload-encoding>US-ASCII</payload-encoding>
405
406         <date>20000508T042651481</date>
407
408         <received >
409             <received-by value="http://foo.com/acc" />
410             <received-date value="20000508T042651481" />
411             <received-id value="123456789" />
412         </received>
413     </params>
414 </envelope>1
415 2
416 --251D738450A171593A1583EB
417 Content-Type: application/text; charset=US-ASCII
418
419 (inform
420     :sender
421         (agent-identifier
422             :name sender@bar.com
423             :addresses (sequence http://bar.com:80/acc))
424     :receiver
425         (set (agent-identifier
426             :name receiver@foo.com
427             :addresses (sequence http://foo.com:80/acc ))))
428     :content-length 12
429     :reply-with task1-003
430     :language fipa-s10
431     :ontology planning-ontology-1
432     :content "
433         (done task1)"
434 --251D738450A171593A1583EB-
435
436

```

<sup>1</sup> CRLF at the end of the XML Envelope.<sup>2</sup> CRLF included in the boundary delimiter at the beginning.

## 436 **5 Informative Annex B — Notes**

### 437 1. Referencing

438 There is no specific reference in the FIPA XML envelope reference to the DTD specified in the in section 2.3,  
439 *Syntax*. This is due to the fact that tests have shown that there is no consistent behaviour of most common parser  
440 in handling a DOCTYPE specification. The most inconvenient fact is that even in the case of non-validation the  
441 parsers are trying to download the DTD from the specified URI.

## 440 6 Informative Annex C — ChangeLog

### 441 6.1 2002/05/26 - version I by FIPA Architecture Board

442 Entire specification: Removed all leading colons (:) from parameter names.

443 Entire specification: Corrected examples.

444 **Page 2, line 90:** **Extended params definition to allow user-defined fields.**

445 **Page 3, line 113:** **Extended agent-identifier definition to allow user-defined fields.**

446 **Page 3, line 130:** **Extended received definition to allow user-defined fields.**

447 **Page 3, line 132:** **Changed type of received-by to url.**

448 **Page 3, line 135:** **Changed type of received-from to url.**

449 Page 4, line 190: Added a rule for prefix string for user-defined fields.

450 **Entire specification:** **Removed all references to the encrypted parameter**

451