

# 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>	SC00085J	<b>Document source</b>	FIPA TC Agent Management
<b>Document status</b>	Standard	<b>Date of this status</b>	2002/12/03
<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.

## 21 **Foreword**

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

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

32 The FIPA specifications are developed through direct involvement of the FIPA membership. The status of a  
33 specification can be either Preliminary, Experimental, Standard, Deprecated or Obsolete. More detail about the process  
34 of specification may be found in the FIPA Document Policy [f-out-00000] and the FIPA Specifications Policy [f-out-  
35 00003]. A complete overview of the FIPA specifications and their current status may be found on the FIPA Web site.

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

39 **Contents**

40	1	Scope.....	1
41	2	XML Envelope Representation.....	2
42	2.1	Component Name .....	2
43	2.2	Mime Type.....	2
44	2.3	Syntax.....	2
45	2.4	Additional Syntax Rules.....	3
46	2.5	Representation of Time .....	4
47	3	References .....	5
48	4	Informative Annex A — Examples .....	6
49	5	Informative Annex B — Notes .....	10
50	6	Informative Annex C — ChangeLog.....	11
51	6.1	2002/11/01 - version I by TC X2S .....	11
52	6.2	2002/12/03 - version J by FIPA Architecture Board .....	11

53 **1 Scope**

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

- 56
- 57 • Syntactic representations of a message envelope in XML form (see [W3Cxml]).
- 58

## 59 2 XML Envelope Representation

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

### 64 2.1 Component Name

65 The name assigned to this component is:

```
66  
67 fipa.mts.env.rep.xml.std  
68
```

### 69 2.2 Mime Type

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

### 73 2.3 Syntax

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

```
75  
76 <!--  
77 Document Type: XML DTD  
78 Document Purpose: Encoding of FIPA ACL message envelopes (as in [FIPA0067]).  
79 See http://www.fipa.org  
80 Last Revised: 2000-08-16  
81 -->  
82  
83 <!ELEMENT envelope ( params+ )>  
84  
85 <!ELEMENT params ( to?,  
86 from?,  
87 comments?,  
88 acl-representation?,  
89 payload-length?,  
90 payload-encoding?,  
91 date?,  
92 encrypted?,  
93 intended-receiver?,  
94 received?,  
95 user-defined* )>  
96  
97 <!ATTLIST params index CDATA #REQUIRED>  
98  
99 <!ELEMENT to ( agent-identifier+ )>  
100  
101 <!ELEMENT from ( agent-identifier )>  
102  
103 <!ELEMENT acl-representation ( #PCDATA )>  
104  
105 <!ELEMENT comments ( #PCDATA )>  
106  
107 <!ELEMENT payload-length ( #PCDATA )>  
108  
109 <!ELEMENT payload-encoding ( #PCDATA )>  
110  
111 <!ELEMENT date ( #PCDATA )>  
112  
113 <!ELEMENT intended-receiver ( agent-identifier+ )>
```

```

114
115
116 <!ELEMENT      agent-identifier      ( name,
117                                     addresses?,
118                                     resolvers?,
119                                     user-defined* )>
120
121 <!ELEMENT      name                   ( #PCDATA )>
122
123 <!ELEMENT      addresses              ( url+ )>
124
125 <!ELEMENT      url                    ( #PCDATA )>
126
127 <!ELEMENT      resolvers              ( agent-identifier+ )>
128
129 <!ELEMENT      received               ( received-by,
130                                     received-from?,
131                                     received-date,
132                                     received-id?,
133                                     received-via?,
134                                     user-defined* )>
135
136 <!ELEMENT      received-by            ( url )>
137
138 <!ELEMENT      received-from          ( url )>
139
140 <!ELEMENT      received-date          EMPTY>
141 <!ATTLIST      received-date          value CDATA #IMPLIED>
142
143 <!ELEMENT      received-id            EMPTY>
144 <!ATTLIST      received-id            value CDATA #IMPLIED>
145
146 <!ELEMENT      received-via           EMPTY>
147 <!ATTLIST      received-via           value CDATA #IMPLIED>
148
149 <!ELEMENT      user-defined           ( #PCDATA )>
150 <!ATTLIST      user-defined           href CDATA #IMPLIED>
151

```

## 152 2.4 Additional Syntax Rules

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

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

173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197

5. The current value of any envelope-level field will be given by the value of the field as it appears in the newest PARAMS element that contains that field.
6. The following pseudo code algorithm may be used to obtain the latest values for each of the envelope parameters:
 

```
EnvelopeWithAllFields := new empty Envelope;

while ((EnvelopeWithAllFields does not contain values for all its fields)
      OR (all PARAMS elements in the sequence have been processed)) {
  // the processor gets the next envelope in the sequence starting with the one
  // with the highest index
  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 contains now the latest values for all its fields set in the envelope.
```
7. User-defined fields in the params, agent-identifier and received parameters must be prefixed with "X-".

198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214

## 2.5 Representation of Time

Time tokens are based on [ISO8601], with extension for relative time and millisecond durations. Time expressions may be absolute, or relative. Relative times are distinguished by the sign character + or - appearing as the first character in the token. If no type designator is given, the local time zone is then used. The type designator for UTC is the character Z; UTC is preferred to prevent time zone ambiguities. Note that years must be encoded in four digits. As an example, 8:30 am on 15th April, 1996 local time would be encoded as:

```
19960415T083000000
```

The same time in UTC would be:

```
19960415T083000000Z
```

while one hour, 15 minutes and 35 milliseconds from now would be:

```
+000000000T011500035
```

### 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, 2000.  
219 <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 IIOP. 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 Times.  
235 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

- 245 1. Here is a simple example of an envelope conforming to the DTD described in Section 2.3:

```

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

```

- 279 2. Here is an example which covers all the aspects described in Section 2.3:

```

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

```

```

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

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

373 MIME-Version: 1.0  
 374 Content-Type: multipart-mixed ;  
 375 boundary="--251D738450A171593A1583EB"  
 376  
 377

378 This is not part of the MIME multipart encoded message.

379 --251D738450A171593A1583EB

380 Content-Type: application/xml

381 <?xml version="1.0"?>

382 <envelope>

383 <params index="1">

384 <to>

385 <agent-identifier>

386 <name>receiver@foo.com</name>

387 <addresses>

388 <url>http://foo.com/acc</url>

389 </addresses>

390 </agent-identifier>

391 </to>

392 <from>

393 <agent-identifier>

394 <name>sender@bar.com</name>

395 <addresses>

396 <url>http://bar.com/acc</url>

397 </addresses>

398 </agent-identifier>

399 </from>

400 <acl-representation>fipa.acl.rep.string.std</acl-representation>

401 <payload-encoding>US-ASCII</payload-encoding>

402 <date>20000508T042651481</date>

403 <received >

404 <received-by value="http://foo.com/acc" />

405 <received-date value="20000508T042651481" />

406 <received-id value="123456789" />

407 </received>

408 </params>

409 </envelope><sup>1</sup>

410 <sup>2</sup>

411 --251D738450A171593A1583EB

412 Content-Type: application/text; charset=US-ASCII

413 (inform

414 :sender

415 (agent-identifier

416 :name sender@bar.com

417 :addresses (sequence http://bar.com:80/acc))

418 :receiver

419 (set (agent-identifier

420 :name receiver@foo.com

421 :addresses (sequence http://foo.com:80/acc )))

422 :content-length 12

<sup>1</sup> CRLF at the end of the XML Envelope.

<sup>2</sup> CRLF included in the boundary delimiter at the beginning.

```
429      :reply-with task1-003
430      :language fipa-s10
431      :ontology planning-ontology-1
432      :content "
433          (done task1)"
434      --251D738450A171593A1583EB-
435
```

## 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. This  
439 is due to the fact that tests have shown that there is no consistent behaviour of most common parser in handling a  
440 DOCTYPE specification. The most inconvenient fact is that even in the case of non-validation the parsers are trying  
441 to download the DTD from the specified URI.  
442

443 **6 Informative Annex C — ChangeLog**

444 **6.1 2002/11/01 - version I by TC X2S**

- 445 Entire document: Removed all : from parameter names
- 446 Entire document: Corrected examples
- 447 **Entire document: Removed all references to the encrypted parameter**
- 448 **Page 2, line 90: Extended params definition to allow user-defined fields**
- 449 **Page 3, line 115: Extended agent-identifier definition to allow user-defined fields**
- 450 **Page 3, line 130: Extended received definition to allow user-defined fields**
- 451 **Page 3, lines 132-133: Changed type of received-by to url**
- 452 **Page 3, lines 135-136: Changed type of received-from to url**
- 453 Page 4, line 190: Added a rule for prefix string for user-defined fields
- 454 **Page 4, line 191: Fixed the definition of relative time**
- 455

456 **6.2 2002/12/03 - version J by FIPA Architecture Board**

- 457 Entire document: Promoted to Standard status
- 458