

FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

FIPA Agent Message Transport Envelope Representation in XML Specification

Document title	FIPA Agent Message Transport Envelope Representation in XML Specification		
Document number	XC000851	Document source	FIPA TC Agent Management
Document status	Experimental	Date of this status	2002/11/01
Supersedes	None		
Contact	fab@fipa.org		
Change history	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 **1 Scope**

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

- 55
- 56 • Syntactic representations of a message envelope in XML form (see [W3Cxml]).

57

58 2 XML Envelope Representation

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

63 2.1 Component Name

64 The name assigned to this component is:

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

68 2.2 Mime Type

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

72 2.3 Syntax

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

```
74 <!--
75 Document Type: XML DTD
76 Document Purpose: Encoding of FIPA ACL message envelopes (as in [FIPA0067]).
77 See http://www.fipa.org
78 Last Revised: 2000-08-16
79 -->
80
81 <!ELEMENT      envelope      ( params+ )>
82
83 <!ELEMENT      params        ( to?,
84                               from?,
85                               comments?,
86                               acl-representation?,
87                               payload-length?,
88                               payload-encoding?,
89                               date?,
90                               encrypted?,
91                               intended-receiver?,
92                               received?,
93                               user-defined* )>
94
95 <!ATTLIST      params        index CDATA #REQUIRED>
96
97 <!ELEMENT      to            ( agent-identifier+ )>
98
99 <!ELEMENT      from          ( agent-identifier )>
100
101 <!ELEMENT      acl-representation ( #PCDATA )>
102
103 <!ELEMENT      comments      ( #PCDATA )>
104
105 <!ELEMENT      payload-length ( #PCDATA )>
106
107 <!ELEMENT      payload-encoding ( #PCDATA )>
108
109 <!ELEMENT      date          ( #PCDATA )>
110
111 <!ELEMENT      intended-receiver ( agent-identifier+ )>
112
```

```

113
114
115 <!ELEMENT      agent-identifier      ( name,
116                                     addresses?,
117                                     resolvers?,
118                                     user-defined* )>
119
120 <!ELEMENT      name                   ( #PCDATA )>
121
122 <!ELEMENT      addresses              ( url+ )>
123
124 <!ELEMENT      url                    ( #PCDATA )>
125
126 <!ELEMENT      resolvers              ( agent-identifier+ )>
127
128 <!ELEMENT      received               ( received-by,
129                                     received-from?,
130                                     received-date,
131                                     received-id?,
132                                     received-via?,
133                                     user-defined* )>
134
135 <!ELEMENT      received-by            ( url )>
136
137 <!ELEMENT      received-from          ( url )>
138
139 <!ELEMENT      received-date          EMPTY>
140 <!ATTLIST      received-date         value CDATA #IMPLIED>
141
142 <!ELEMENT      received-id            EMPTY>
143 <!ATTLIST      received-id           value CDATA #IMPLIED>
144
145 <!ELEMENT      received-via          EMPTY>
146 <!ATTLIST      received-via         value CDATA #IMPLIED>
147
148 <!ELEMENT      user-defined           ( #PCDATA )>
149 <!ATTLIST      user-defined          href CDATA #IMPLIED>
150

```

151 2.4 Additional Syntax Rules

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

- 153
- 154 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.
- 155
- 156
- 157
- 158
- 159 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.
- 160
- 161
- 162
- 163 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.
- 164
- 165
- 166
- 167
- 168
- 169
- 170 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.
- 171

172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196

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-".

197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213

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
```

214 3 References

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

243 4 Informative Annex A — Examples

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

```

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

```

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

```

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

```

```

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

```

```
368     </envelope>
```

- 369
370 3. Here is an example which also includes the MIME multipart encapsulation which might be used over HTTP (see
371 [FIPA00084]):

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

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

```
377 --251D738450A171593A1583EB
378 Content-Type: application/xml
379
```

```
380
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
401     <acl-representation>fipa.acl.rep.string.std</acl-representation>
402
403     <payload-encoding>US-ASCII</payload-encoding>
404
405     <date>20000508T042651481</date>
406
407     <received >
408       <received-by value="http://foo.com/acc" />
409       <received-date value="20000508T042651481" />
410       <received-id value="123456789" />
411     </received>
412   </params>
413 </envelope>1
414 2
415 --251D738450A171593A1583EB
416 Content-Type: application/text; charset=US-ASCII
417
418 (inform
419   :sender
420     (agent-identifier
421       :name sender@bar.com
422       :addresses (sequence http://bar.com:80/acc))
423   :receiver
424     (set (agent-identifier
425       :name receiver@foo.com
426       :addresses (sequence http://foo.com:80/acc )))
427   :content-length 12
```

¹ CRLF at the end of the XML Envelope.

² CRLF included in the boundary delimiter at the beginning.

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

435 **5 Informative Annex B — Notes**

436 1. Referencing

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

442 6 Informative Annex C — ChangeLog

443 6.1 2002/11/01 - version I by TC X2S

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