1

FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

2

3

4

FIPA Messaging Interoperability Service Specification

5

6

7

8

Document title	FIPA Messaging Interoperability Service Specification		
Document number	PC00093A	Document source	FIPA TC Gateways
Document status	Preliminary	Date of this status	2001/08/10
Supersedes	None		
Contact	gateways@fipa.org		
Change history			
2001/07/25	Initial Version		
2001/08/10	Line numbering added		

9

10

11

12

13

14

15

16

17 © 2001 Foundation for Intelligent Physical Agents - http://www.fipa.org/

18 *Geneva, Switzerland*

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.

73 2   Overview

74 The FIPA Messaging Interoperability Service (FIPA-MIS) provides a means for converting between Message Transport
75 Protocols (MTPs) and between concrete encodings of FIPA-message parts. FIPA-MIS can be used where direct end-
76 to-end interoperability is impossible, impractical or undesirable. Direct end-to-end interoperability is impossible when
77 communicating platforms/agents do not support any common message transport protocol or encoding of FIPA-
78 message components, for example. Direct end-to-end interoperability may be impractical when communicating over a
79 slow wireless link with a peer in the fixed network that does not support any message transport protocol suitable for
80 wireless links.
81

82 2.1   Reference Model

83 The reference model for FMIS comprises four levels (see *Figure 1*):
84

85 1. The Message Transport Protocol Gateway (MTP-GW) is used to translate between Message Transport Protocols.
86   For example, the Message Transport Protocol Gateway may translate between `fipa.mts.mtp.iiop.std` and
87   `fipa.mts.mtp.wap.std`.
88

89 2. The Message Envelope Encoding Gateway (ENV-GW) is used to translate between Message Envelope encodings.
90   For example, the Message Envelope Encoding Gateway may translate between `fipa.mts.env.rep.xml.std`
91   and `fipa.mts.env.rep.bitefficient.std`.
92

93 3. The ACL Encoding Gateway (ACL-GW) is used to translate between ACL encodings. For example, the ACL
94   Encoding Gateway may translate between `fipa.acl.rep.xml.std` and `fipa.acl.rep.bitefficient.std`.
95

96 4. The Content Language Encoding Gateway (CL-GW) is used to translate between Content Language encodings.
97   Note that the current specification does not allow conversion between *different* content languages, only between
98   *different encodings* of the same content language[1]. However, if this kind of functionality is needed, in can be added
99   easily to the gateway specification. How such a translation is actually performed is an application implementation
100   issue, and hence is out of scope.
101

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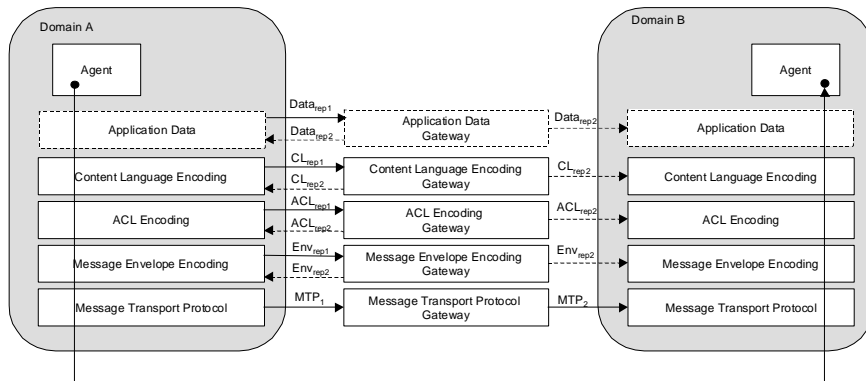104 specification.

Figure 1: FIPA Messaging Interoperability Service Reference Model

105
106
107

---

[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.

131 # 4 Messaging Interoperability Service Ontology

132 ## 4.1 Object Descriptions

133 This section describes a set of frames, that represent the classes of objects in the domain of discourse within the
134 framework of the `FIPA-MIS` ontology.
135
136 The following terms are used to describe the objects of the domain:
137
138   **Frame**. This is the mandatory name of this entity, that must be used to represent each instance of this class.
139
140   **Ontology**. This is the name of the ontology, whose domain of discourse includes the parameters described in the
141   table.
142
143   **Parameter**. This is the mandatory name of a parameter of this frame.
144
145   **Description**. This is a natural language description of the semantics of each parameter.
146
147   **Presence**. This indicates whether each parameter is mandatory or optional.
148
149   **Type**. This is the type of the values of the parameter: Integer, Word, String, URL, Term, Set or Sequence.
150
151   **Reserved Values**. This is a list of FIPA-defined constants that can assume values for this parameter.
152

153 ### 4.1.1 Translation Identifier

154 This type of object represents the unique identification for the incoming message translation.
155

Frame Ontology	`translation-id` `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`	

156

157 ## 4.2 Function Descriptions

158 The following tables define usage and semantics of the functions that are part of the `FIPA-MIS` ontology.
159
160 The following terms are used to describe the functions of the `FIPA-MIS` domain:
161
162   **Function**. This is the symbol that identifies the function in the ontology.
163
164   **Ontology**. This is the name of the ontology, whose domain of discourse includes the function described in the
165   table.
166
167   **Supported by**. This is the type of agent that supports this function.
168
169   **Description**. This is a natural language description of the semantics of the function.
170
171   **Domain**. This indicates the domain over which the function is defined. The arguments passed to the function must
172   belong to the set identified by the domain.

173

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

176

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

179

180   **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

181

182   **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

183

184   **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

185
186

---

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

186   **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

187

188   **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

189

190   **4.3   Exceptions**

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

192

193   **4.3.1   Not Understood Exception Propositions**

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

196

197   **4.3.2   Refusal Exception Propositions**

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

200

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

201
202

202

203   **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.

204

205

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)))))
```

256 **6   References**

257 [FIPA00023]   FIPA Agent Management Specification. Foundation for Intelligent Physical Agents, 2000.
258   `http://www.fipa.org/specs/fipa00023/`
259 [FIPA00067]   FIPA Agent Message Transport Service Specification. Foundation for Intelligent Physical Agents, 2000.
260   `http://www.fipa.org/specs/fipa00067/`

261

261 # 7   Informative Annex A — Examples

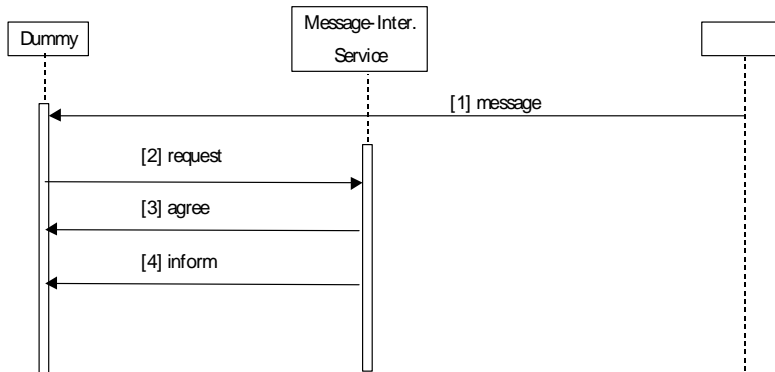262 ## 7.1   Transformation Encoding Request

Figure 2: Transformation of message-component encoding

263 This example shows how an agent requests the Messaging Interoperability Service to transform a message component
264 from one encoding to another. The message flow is illustrated in *Figure 2*.
265
266 1. Message [1]: The agent *dummy* receives a message and wants to transform the ACL-encoding of the message.
267
268 2. Message [2] `request`: The agent *dummy* sends the transform request to the Messaging Interoperability Service.
269   The request contains the message-component to be transformed and the requested new encoding representation.
270

```
271 (request
272   :sender
273     (agent-identifier
274       :name dummy
275       :addresses (sequence http://helluli.com/acc))
276   :receiver (set
277     (agent-identifier
278       :name fipa-messaging-interoperability-service
279       :addresses (sequence http://fmis.com/acc)))
280   :ontology FIPA-MIS
281   :language fipa-sl0
282   :protocol fipa-request
283   :content
284     (action
285       (agent-identifier
286         :name fipa-messaging-interoperability-service)
287       (transform
288         (message-component (request …) )
289         (encoding-representation
290           :name fipa.acl.rep.bitefficient.std))))
291
292
```

```
292    3.  Message [3] agree: The Messaging  Interoperability Service agrees to perform the transformation:
293
294    (agree
295      :sender
296        (agent-identifier
297          :name fipa-messaging-interoperability-service
298          :addresses (sequence http://fmis.com/acc))
299      :receiver (set
300        (agent-identifier
301          :name dummy
302          :addresses (sequence http://helluli.com/acc)))
303      :ontology FIPA-MIS
304      :language fipa-sl0
305      :protocol fipa-request
306      :content
307        ((action
308          (agent-identifier
309             :name fipa-messaging-interoperability-service)
310          (transform
311            (message-component (request …) )
312            (encoding-representation
313              :name fipa.acl.rep.bitefficient.std)))
314           true))
315
316    4.  Message [4] inform: The Messaging Interoperability Service returns the encoded message component to the
317        agent.
318
319    (inform
320      :sender
321        (agent-identifier
322          :name fipa-messaging-interoperability-service
323          :addresses (sequence http://fmis.com/acc))
324      :receiver (set
325        (agent-identifier
326          :name dummy
327          :addresses (sequence http://helluli.com/acc)))
328      :ontology FIPA-MIS
329      :language fipa-sl0
330      :protocol fipa-request
331      :content
332        (result
333          (action
334            (agent-identifier
335               :name fipa-messaging-interoperability-service)
336            (transform
337              (message-component (request …) )
338            (encoding-representation
339                 :name fipa.acl.rep.bitefficient.std)))
340        (message-component 0xfa… )))
341
```

341

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)))
```

390 **7.3　Receiver initialised transformations**

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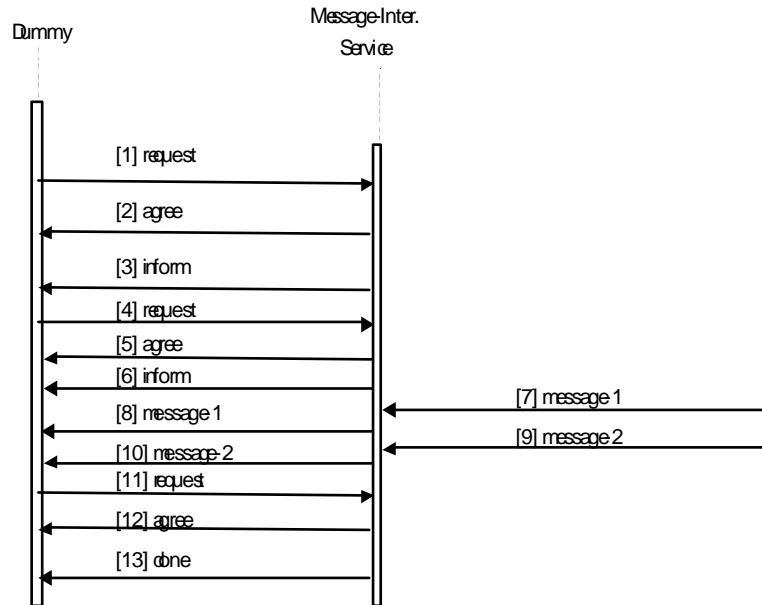392 delivery to the agent.

Figure 3: Receiver Initialised Transformations

393

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

396
```
397 (request
398   :sender
399     (agent-identifier
400       :name dummy
401       :addresses (sequence http://campa.com/acc))
402   :receiver (set
403     (agent-identifier
404       :name fipa-messaging-interoperability-service
405       :addresses (sequence http://fmis.com/acc)))
406   :ontology FIPA-MIS
407   :language fipa-sl0
408   :protocol fipa-request
409   :content
410     (action
411       (agent-identifier
412         :name fipa-messaging-interoperability-service)
413       (available-encodings)))
```
414

414
415   2.   Message [2] `agree`: The Messaging Interoperability Service agrees to deliver the list.
416
417   3.   Message [3] `inform`: The Messaging Interoperability Service sends the list:
418
```
419 (inform
420   :sender
421    (agent-identifier
422       :name fipa-messaging-interoperability-service
423       :addresses (sequence http://fmis.com/acc))
424   :receiver (set
425     (agent-identifier
426       :name dummy
427       :addresses (sequence http://campa.com/acc)))
428   :ontology FIPA-MIS
429   :language fipa-sl0
430   :protocol fipa-request
431   :content
432     (result
433       (action
434         (agent-identifier
435           :name fipa-messaging-interoperability-service)
436       (available-encodings))
437     (gateway-description
438       :acl-translation
439         (set
440           (acl-gw-description
441             :from fipa.acl.rep.bitefficient.std
442             :to (set fipa.acl.rep.string.std fipa.acl.rep.xml.std))
443           (acl-gw-description
444             :from fipa.acl.rep.string.std
445             :to (set fipa.acl.rep.bitefficient.std))))))
```
446
447   4.   Message [4] `request`: The agent *dummy* requests to the Messaging Interoperability Service to transform
448   messages to the `fipa.acl.rep.bitefficient.std` encoding before delivering them to the agent *dummy*:
449
```
450 (request
451   :sender
452     (agent-identifier
453       :name dummy
454       :addresses (sequence http://campa.com/acc))
455   :receiver (set
456     (agent-identifier
457       :name fipa-messaging-interoperability-service
458       :addresses (sequence http://fmis.com/acc)))
459   :ontology FIPA-MIS
460   :language fipa-sl0
461   :protocol fipa-request
462   :content
463     (action (agent-identifier :name fipa-messaging-interoperability-service)
464       (incoming-translation
465         (sequence
466           (gateway-behaviour
467             :acl fipa.acl.rep.bitefficient.std)))))
```
468
469   5.   Message [5] `agree`: The Messaging Interoperability Service agrees.
470
471

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

472

```
473   (inform
474     :sender
475      (agent-identifier
476         :name fipa-messaging-interoperability-service
477         :addresses (sequence http://fmis.com/acc))
478     :receiver (set
479      (agent-identifier
480         :name dummy
481         :addresses (sequence http://campa.com/acc)))
482     :ontology FIPA-MIS
483     :language fipa-sl0
484     :protocol fipa-request
485     :content
486      (result
487       (action (agent-identifier :name fipa-messaging-interoperability-service)
488        (incoming-translation
489          (sequence
490            (gateway-behaviour
491               :acl fipa.acl.rep.bitefficient.std))))
492       (translation-id :id id1)))
```

493

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

496

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

498

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

500

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

503

```
504   (request
505     :sender
506       (agent-identifier
507         :name dummy
508         :addresses (sequence http://campa.com/acc))
509     :receiver (set
510      (agent-identifier
511         :name fipa-messaging-interoperability-service
512         :addresses (sequence http://fmis.com/acc)))
513     :ontology FIPA-MIS
514     :language fipa-sl0
515     :protocol fipa-request
516     :content
517      (action (agent-identifier :name fipa-messaging-interoperability-service)
518        (received-translated-cancel
519          (translation-id :id id1))))
```

520

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

522

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

525

526