FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

FIPA Personal Assistant Specification

Document title	FIPA Personal Assistant Specification		
Document number	XC00083B	Document source	FIPA Architecture Board
Document status	Experimental	Date of this status	2001/08/10
Supersedes	FIPA00022		
Contact	fab@fipa.org		
Change history			
2000/10/17	Approved for Experimental		
2001/08/10	Line numbering added		

© 2000 Foundation for Intelligent Physical Agents - http://www.fipa.org/

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

One central class of agents is that of a Personal Assistant (PA). A PA is a software agent that acts semi-autonomously for and on behalf of a user, modelling the interests of the user and providing services to the user or other users and PAs as and when required. A PA is like a secretary, it accomplishes routine support tasks to allow the user to concentrate on their real job. It is unobtrusive but ready when needed and rich in knowledge about user and their areas of work.

The notion of a PA is very broad. There are many internal and external functions and services that can and will be used to provide and extend a PA's basic functionalities. In fact, such encapsulation of new services is a critical requirement where the interoperability of a PA's functions and services is desirable. The use of agent technology to support a PA helps in achieving this requirement.

Examples of such functions and services include:

Managing a user's diaries,

Filtering and sorting email,

Managing a user's desktop environment,

Managing a user's activities, plans and tasks,

Locating and delivering multimedia information,

Recommending entertainment,

Purchasing desired items, and,

Planning travel.

While this specification focuses on the generic requirements for a PA application, other FIPA specifications, such as [FIPA00080] and [FIPA00081], also include the notion of personal assistance for specific applications. The Personal Travel Assistant (PTA) service defined in [FIPA00080] is different from a PA service since the PTA is an integrated system which spans a particular market segment of the electronic marketplace; the PA provides a user-oriented front-end to a wide variety of different services, one of which may be a PTA.

97 ## **2 General Analysis**

98 In general, a PA comprises (see *Figure 1*):

99
100 Intelligence and associated capabilities such as rationality (reasoning and planning) and adaptability/learning,
101
102 Knowledge including facts, rules and adapted/learned knowledge for and about an end-user,
103
104 Interaction capabilities and facilities with the user, other agents and software and hardware services and functions,
105 and,
106
107 The services and functions and their procedures for the agent to work with them.
108
109 The scope of this composite is limited to the tasks which are given by the user as goals and preferences for behaviour.
110 Other agents will also exist and interact with the PA, but such other agents will not tend to represent particular user's
111 preferences or access authority and other differentiators.
112
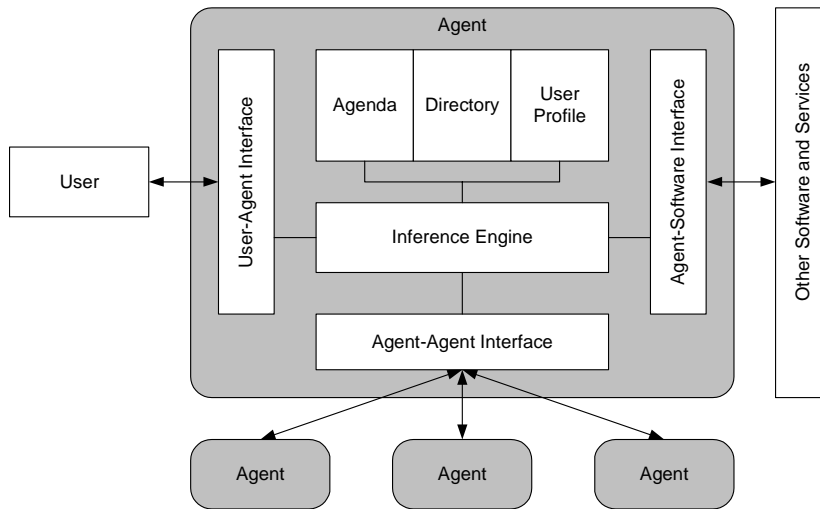
115 **Figure 1:** Personal Assistant Reference Model

117 The reference model include the following interfaces and protocols of interaction that are candidates for standardisation:
118
119 User-agent dialogue,
120
121 Multi-modal user-agent interface,
122
123 Agent-agent communication interface,
124
125 Protocols for agent-agent interaction,
126
127 Agent-software interfaces,
128
129 Agent-software communication protocols,
130
131 Agent-function interfaces, and,
132
133 Function interoperability interfaces and mechanisms.
134

135 Multi-modality is the ultimate goal for human-agent interfaces. As a user interacts with a real PA, he/she can speak
136 face-to-face with the PA or over the phone. Obviously, unconstrained natural language comprehension is also
137 desirable. However, as first steps toward this general goal, multi-modal interaction is taken not as requirement for all
138 agents to support all modalities, but merely that any single application should be able to select the one or more
139 modalities required for the application. The composition of the PA should be media-independent in order to allow for this
140 choice and otherwise provide the multi-modal conversions required, such as converting text to speech in order to pass
141 information over the phone.
142

143 ## 2.1   Directory Services

144 One of the basic functions of a PA is the management of the user's directory which includes telephone numbers,
145 addresses and personal and useful information about them. This information facilitates responses the PA may provide
146 to user's needs in an intelligent way, based on the context of the request. For example, if the user asks to call an
147 organisation and the PA, by reference to the opening and closing times, notices that the call cannot be made then it will
148 suggest alternative actions by inferring the user's possible intentions on the basis of the services provided by the
149 organisation. A request for a call to a specific travel agency out of opening hours may result in the PA suggesting
150 contacting a 24 hour call centre of an airline company.
151

152 ## 2.2   Meeting Scheduling Services

153 The PA includes a calendar facility, that among other things can be used for scheduling of meetings and negotiation
154 with users and other agents. This functionality includes:
155
156   Identifying a mutually appropriate time for all participants to attend a meeting,
157
158   Reserving an appropriate venue for the meeting,
159
160   Organising associated facilities,
161
162   Issuing reminders to all participants,
163
164   Handling any problems which might arise at a later date, and,
165
166   Cancelling meetings.
167

168 ## 2.3   Information Management Services

169 This is a very large and nebulous set of tasks but also addresses one of the most critical needs for intelligent personal
170 assistants. Most professionals are now inundated with too many sources of information and generally this is called
171 *information overload*. An agent can semi-autonomously filter, sort or otherwise respond to all these sources to help off-
172 load some of the more mundane tasks these professionals now must do themselves. Such task include:
173
174   Email and news filtering,
175
176   Sorting and prioritising received information, and,
177
178   Automatically responding or forwarding information to another user.
179
180 A key aspect of such information management is not just filtering out the low priority information, but also providing the
181 timely delivery of high priority items; anywhere, anytime, anyhow. Such delivery is dependent on the user's location,
182 media/equipment limitations and preferences. For instance, an agent can be instructed to deliver important email to an
183 end-user even if the user only has a mobile phone by converting the text to speech. Of course, this same text to speech
184 delivery of email over a mobile phone can be applied to any text-based information source such as NNTP news, stock
185 quotes, etc. Furthermore, given the cost of mobile phone connectivity, other technologies such as text summarisation

186 can be employed, for the most efficient delivery. The provision of such summarisation and media to media
187 transformation could be provided, for example, via external services.
188
189 Even under the most constrained situations, such as the user only having a pager, a PA can at least notify the user
190 about the existence and accessibility of an important new multimedia document. Though the pager device cannot
191 deliver the information, the PA can notify the user of the appropriate equipment in the locality of the user that is
192 available where the multimedia document could be sent.
193
194 A less well-developed but equally important aspect of information management is the personal storage and retrieval of
195 information. Even personal computer storage is becoming difficult to manage since files are often duplicated, directory
196 structures are haphazard and the file systems themselves does not provide rich indexing and content search facilities.
197 Here, a PA can be asked to file and retrieve documents or even isolated bits of information, much like a personal
198 secretary manages paper-based documents in the office.
199

2.4   Travel Planning Service

201 A PA can assist in planning the user's trip by interacting with the user, other agents and external directory services and
202 thus providing an appropriate plan of an intended trip and other guidance services (see [FIPA00079] for more
203 information). In particular, the PA may assume much of the functionality of the PTA agent.
204

205

3   Scenario

The chosen scenario to illustrate the PA application is that of arranging meetings among several participants, located across companies and using different calendar management systems. This has been chosen for several reasons:

The scenario is instantly applicable to real-life situations,

Current solutions are unwieldy and proprietary, making integration across networks and other services difficult,

The scenario can be easily extended to incorporate further services, and,

This class of application has been widely studied, with many agent-based approaches being developed (for example, see [Kozieriok1993]).

The scenario lends itself well to agent technology, due to the need for:

User profiling,

Integration of heterogeneous software,

Action on a user's behalf (semi-autonomy), and,

Local control (in particular, of the user's calendar).

The selected service is an integration of meeting scheduling and travel assistance. The user asks the agent to set up a meeting with several participants and because the meeting may involve travelling for some of the participants, travel planning forms part of the meeting scheduling.

3.1   Architecture

In this test application, the PA provides a single unified interface to the user, and is represented as a single agent, even though it may actually be composed of several agents (see *Figure 2*). Each user is represented by a PA and they negotiate with each other to find an adequate meeting place and time. They interact with Travel Broker Agents (TBAs) to find out about travel schedules, etc. (see [FIPA00079]).
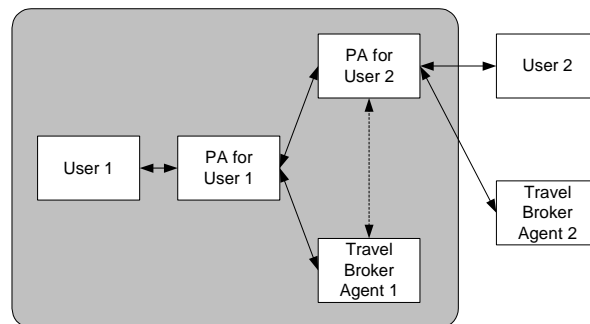

Figure 2: Agent Interactions in the Personal Assistant Application Scenario

The PA may make use of several existing programs to perform its tasks. A possible set of relevant software for this test application is shown in *Figure 3*.
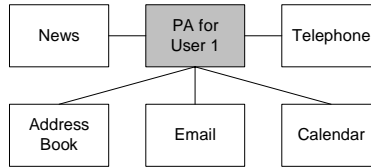
245
246
247   **Figure 3:** Integration of External Software
248
249 This specification defines a set of scenarios based on one particular aspect of the PA role; that of arranging
250 appointments for activities, such as meetings, which involve a number of participants. The PAs of the participants must
251 cooperate to ensure that the meeting is arranged for a time and place which is mutually acceptable to all their users.
252
253   **Basic scenario**
254   User 1 is chairman of a committee and asks his PA to arrange a committee meeting.
255
256   **Sub-scenario 1**
257   User 1 suggests a specific time, duration and location for the meeting. The simplest case is that all participants are
258   required to attend the meeting.
259
260   **Sub-scenario 2**
261   User 1 suggests a range of possible times to start the meeting. The duration and location of the meeting remain
262   specific.
263
264   **Sub-scenario 3**
265   User 1 suggests a range of possible times to start the meeting as well as a set of locations at which the meeting
266   can take place.
267
268 In the scenarios, only those participants which have PAs will be considered in the agent-based negotiation of the
269 meeting. Other possible interfaces (directly between personal agent and another human) will not be required.
270
271 It is up to the individual PA application to associate PAs with their users, that is, if the initiator enters a list of participants
272 to the meeting, the initiator's agent needs to be able to communicate with the PAs of the participants. The above sub-
273 scenarios are in order of increasing sophistication and difficulty of implementation. Further scenarios, which are not
274 directly under consideration in this specification, could be achieved by allowing negotiation about the duration and
275 actual participants in the meeting.
276
277 A field trial prototype should demonstrate cooperative meeting arrangement involving several PAs. It should implement
278 at least the functionality required by the simplest sub-scenario (1) and preferably more. One or more of the PAs should
279 make use of travel planning services provided by a PTA field trial system. It is preferable that interoperability of at least
280 two independently-designed and implemented types of PA be demonstrated.
281

282 ## 3.2   Human-Agent Interaction
283 Humans may be viewed as agents in their own right and that their interaction with software and agents may also be
284 achieved using ACL. However, since humans do not speak ACL directly, some translation service is required to present
285 a user-friendly interface. This may be accomplished by using a special user interface agent as illustrated in *Figure 4*.
286
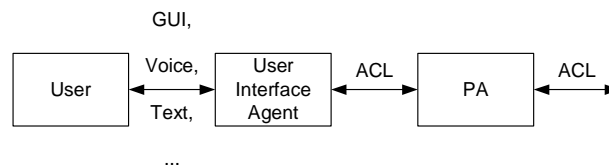
287
288
289   **Figure 4:** Human-Agent Interaction Via a User Interface Agent
290

291   Thus, the following interactions between users and their PAs could be:

292

293   User to PA: Give task to arrange meeting (including reporting requirements),

294

295   PA to User: Give progress status of task (at least success or failure),

296

297   PA to User: Require permission to commit to the meeting,

298

299   User to PA: Give permission to commit to the meeting,

300

301   PA to User: Inform of commitment made, and,

302

303   User to PA: Give degree of delegation authority.

304

305

305 # 4   Personal Assistant Ontology

306 ## 4.1   Object Descriptions

307 This section describes a set of frames, that represent the classes of objects in the domain of discourse within the
308 framework of the `FIPA-PA` ontology.
309
310 The following terms are used to describe the objects of the domain:
311
312   **Frame**. This is the mandatory name of this entity, that must be used to represent each instance of this class.
313
314   **Ontology**. This is the name of the ontology, whose domain of discourse includes the parameters described in the
315   table.
316
317   **Parameter**. This is the mandatory name of a parameter of this frame.
318
319   **Description**. This is a natural language description of the semantics of each parameter.
320
321   **Presence**. This indicates whether each parameter is mandatory or optional.
322
323   **Type**. This is the type of the values of the parameter: Integer, Word, String, URL, Term, Set or Sequence.
324
325   **Reserved Values**. This is a list of FIPA-defined constants that can assume values for this parameter.
326

327 ### 4.1.1   Meeting Description

328 This type of object represents the description of a meeting event. The syntax and format of the `:details` parameter is
329 given in [RFC2445].
330

Frame **Ontology**	`meeting-description` `FIPA-PA`			
Parameter	**Description**	**Presence**	**Type**	**Reserved Values**
`identifier`	The identifier of the meeting.	Mandatory	`String`	
`user`	The user organising the meeting.	Optional	`String`	
`details`	The details of the meeting.	Mandatory	`String`	See [RFC2445]

331

332 ## 4.2   Function Descriptions

333 The following tables define usage and semantics of the functions that are part of the `FIPA-PA` ontology.
334
335 The following terms are used to describe the functions of the `FIPA-PA` domain:
336
337   **Function**. This is the symbol that identifies the function in the ontology.
338
339   **Ontology**. This is the name of the ontology, whose domain of discourse includes the function described in the
340   table.
341
342   **Supported by**. This is the type of agent that supports this function.
343
344   **Description**. This is a natural language description of the semantics of the function.
345

346   **Domain**. This indicates the domain over which the function is defined. The arguments passed to the function must
347   belong to the set identified by the domain.
348
349   **Range**. This indicates the range to which the function maps the symbols of the domain. The result of the function is
350   a symbol belonging to the set identified by the range.
351
352   **Arity**. This indicates the number of arguments that a function takes. If a function can take an arbitrary number of
353   arguments, then its arity is undefined.
354

355   **4.2.1   Negotiate a Meeting**

Function	`meet`
Ontology	`FIPA-PA`
Supported by	PA
Description	The execution of this function has the effect asking the PA to negotiate a meeting between the initiator and the attendees using the `FIPA-Contract-Net` interaction protocol (see [FIPA00029]). Once the details of a meeting have been decided, then it can be scheduled (see section *4.2.2, Schedule a Meeting*). The first argument contains the details of the meeting to schedule, the second argument contains a list of time frames at which it is proposed that the meeting is scheduled and the third object is the duration of the meeting.
Domain	`meeting-description`, Sequence of `DateTime` (see [FIPA00070]), `Integer`
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	3

356

357   **4.2.2   Schedule a Meeting**

Function	`schedule`
Ontology	`FIPA-PA`
Supported by	PA
Description	The execution of this function has the effect asking the PA to schedule a meeting between the initiator and the attendees. The first argument contains the details of the meeting to schedule, the second argument contains the time frame of the meeting.
Domain	`meeting-description`, `DateTime`
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	2

358

359   **4.2.3   Participate in a Meeting**

Function	`participate`
Ontology	`FIPA-PA`
Supported by	PA
Description	The execution of this function has the effect of asking the PA to participate in a meeting. The first argument contains the name of the participant and the second argument contains the details of the meeting to schedule.
Domain	`String`, `meeting-description`
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	2

360
361

361   **4.2.4   Schedule a Trip**

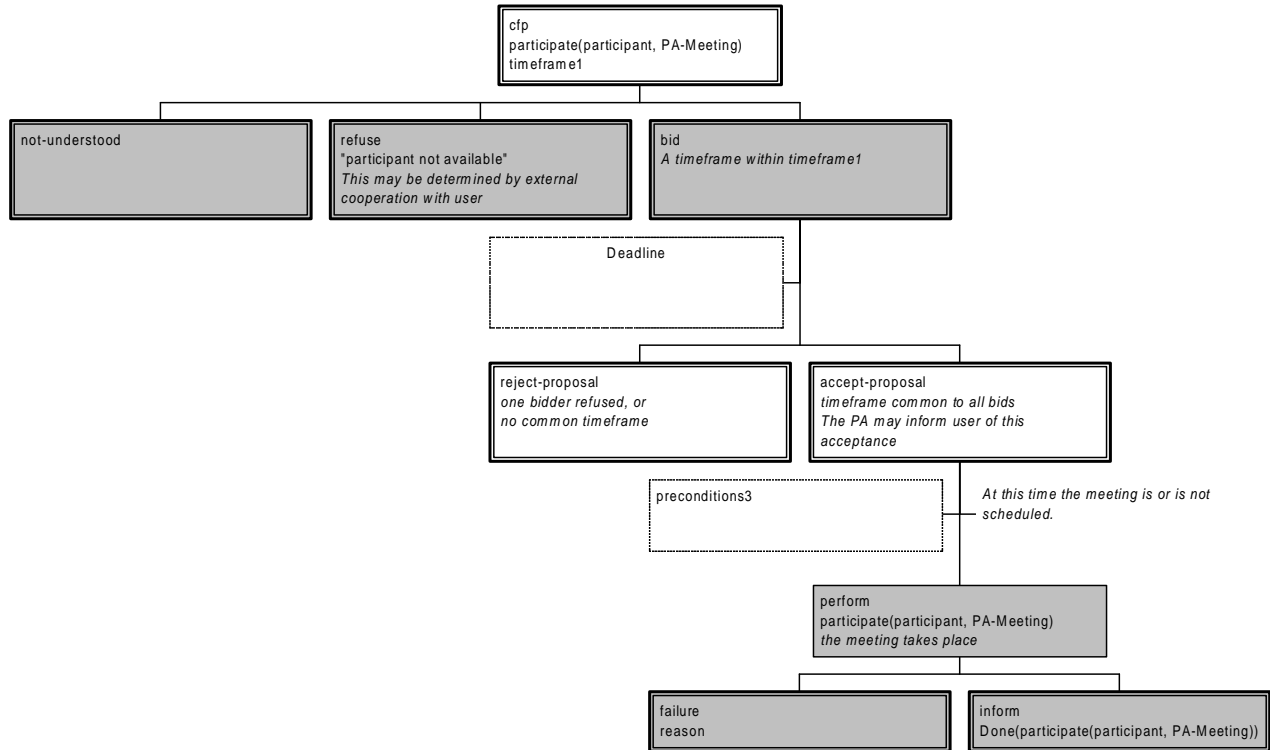| Function	`travel`
Ontology	`FIPA-PA`
Supported by	PA
Description	The execution of this function has the effect of asking the PA to schedule travel for the initiator. The first argument contains the details of the trip to schedule, the second argument contains a list of time frames at which it is proposed that the trip is scheduled and the third object is the duration of the trip.
Domain	`trip-summary` (see [FIPA00080]), Sequence of `DateTime, Integer`
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	3

362

363

364

364 **4.3 Interaction Protocols**

365 **4.3.1 Negotiating Meeting Details**

366 This normative interaction protocol (see *Figure 5*) is used for negotiation of meetings (more accurately, negotiation
367 about the action of participating in the meetings) among the PAs. Whereas the intent of the original contract net protocol
368 (see [FIPA00029]) was to select the best bid from a set of submitted bids, here the `accept-proposal` message is
369 used only if all bids (that is, time frames to attend the meeting) have a time frame in common, which becomes the time
370 frame that is chosen. As the bids of contract net imply firm commitment, the PAs need to check with their users and the
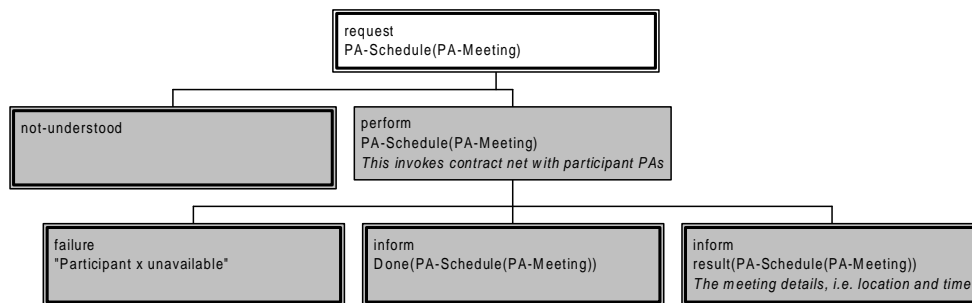371 initiator agent.

372

```
                              cfp
                              participate(participant, PA-Meeting)
                              timeframe1

  not-understood          refuse                             bid
                          "participant not available"        A timeframe within timeframe1
                          This may be determined by external
                          cooperation with user

                                         Deadline


                              reject-proposal              accept-proposal
                              one bidder refused, or        timeframe common to all bids
                              no common timeframe           The PA may inform user of this
                                                            acceptance

                                   preconditions3           At this time the meeting is or is not
                                                            scheduled.


                                         perform
                                         participate(participant, PA-Meeting)
                                         the meeting takes place

                              failure                       inform
                              reason                        Done(participate(participant, PA-Meeting))
```

373
374
375 **Figure 5:** Negotiating Meeting Details Using the Contract Net Interaction Protocol
376

377 **4.3.2 Scheduling a Meeting**

378 This informative interaction protocol (see *Figure 6*) is used between the user and their PA and is modelled using an
379 `order` protocol that can be used to order the initial negotiation of a meeting.

380

```
                              request
                              PA-Schedule(PA-Meeting)

  not-understood          perform
                          PA-Schedule(PA-Meeting)
                          This invokes contract net with participant PAs

            failure                    inform                      inform
            "Participant x unavailable" Done(PA-Schedule(PA-Meeting))  result(PA-Schedule(PA-Meeting))
                                                                   The meeting details, i.e. location and time
```

381
382
383 **Figure 6:** Scheduling a Meeting Using an Order Interaction Protocol
384
385

385 **4.3.3 Confirming a Meeting**

386 This informative interaction protocol (see *Figure 7*) is used between the user and their PA and is modelled using the
387 `FIPA-Request-When` interaction protocol (see [FIPA00028]) to confirm the availability of the user at a suggest
388 meeting time.

389
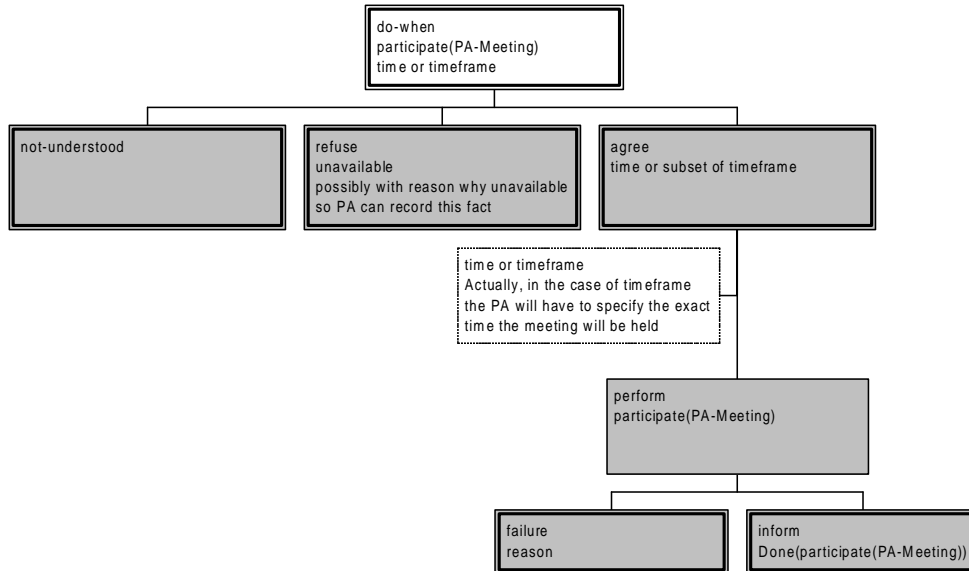
390
391
392 **Figure 7:** Confirming a Meeting using the Request-When Interaction Protocol
393
394

394   **5   Examples**

395   1. This example will schedule an hour long meeting with colleagues for a specific time on a given day:

396

```
397  (cfp
398    :sender
399      (agent-identifier
400        :name UA-John)
401    :receiver (set
402      (agent-identifier
403        :name PA-Peter))
404    :ontology FIPA-PA
405    :protocol FIPA-ContractNet
406    :reply-with Response123
407    :reply-by 19960315T083000000
408    :conversation-id UA-John345
409    :content (meet
410      (meeting-description
411        :identifier John123
412        :agent "Joe Bloggs"
413        :details (
414          BEGIN: VCALENDAR
415          VERSION: 1.0
416          BEGIN: VEVENT
417          UID: 123
418          SUMMARY: Test
419          ATTENDEE: Joe Bloggs
420          ATTENDEE; ROLE=ORGANIZER: John Doe
421          LOCATION: Area 51
422          CATEGORIES: X-FIPA-Test
423          DESCRIPTION: This is just a test meeting. Please do not attempt to attend it.
424          SEQUENCE: 0
425          PRIORITY: 0
426          DTSTART: X-FIPA-UnderNegotiation
427          DTEND: X-FIPA-UnderNegotiation
428          STATUS: UNDER NEGOTIATION
429          X-FIPA-ORGANIZER: John Doe
430          END: VEVENT
431          END: VCALENDAR
432        )
433      (sequence 19960415T083000000)
434      60))
```

435

436

6   References

436

437 [FIPA00023]   FIPA Agent Management Specification. Foundation for Intelligent Physical Agents, 2000.
438   `http://www.fipa.org/specs/fipa00023/`
439 [FIPA00028]   FIPA Request When Interaction Protocol Specification. Foundation for Intelligent Physical Agents,
440   2000.
441   `http://www.fipa.org/specs/fipa00028/`
442 [FIPA00029]   FIPA Contract Net Interaction Protocol Specification. Foundation for Intelligent Physical Agents, 2000.
443   `http://www.fipa.org/specs/fipa00029/`
444 [FIPA00070]   FIPA ACL Message Representation in String Specification. Foundation for Intelligent Physical Agents,
445   2000.
446   `http://www.fipa.org/specs/fipa00070/`
447 [FIPA00079]   FIPA Agent Software Integration Specification. Foundation for Intelligent Physical Agents, 2000.
448   `http://www.fipa.org/specs/fipa00079/`
449 [FIPA00080]   FIPA Personal Travel Assistance Specification. Foundation for Intelligent Physical Agents, 2000.
450   `http://www.fipa.org/specs/fipa00080/`
451 [FIPA00081]   FIPA Audio/Visual Entertainment and Broadcasting Specification. Foundation for Intelligent Physical
452   Agents, 2000.
453   `http://www.fipa.org/specs/fipa00081/`
454 [Kozieriok1993] A Learning Interface Agent for Scheduling Meetings, Kozieriok, R. and Maes, P. International
455   Workshop on Intelligent User Interfaces, AMC-SIGCHI, AMC Press, Florida, 1993.
456 [RFC2445]   Internet Calendaring and Scheduling Core Object Specification (iCalendar). Request for Comments,
457   1998.
458   `http://www.ietf.org/rfc/rfc2445.txt`