FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

FIPA Contract Net Interaction Protocol Specification

Document title	FIPA Contract Net Interaction Protocol Specification		
Document number	XC00029G	**Document source**	FIPA TC Communication
Document status	Experimental	**Date of this status**	2002/11/01
Supersedes	None		
Contact	fab@fipa.org		
Change history	See *Informative Annex A — ChangeLog*		

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 where appropriate.

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 Document Policy [f-out-00000] and the FIPA Specifications Policy [f-out-00003]. A complete overview of the FIPA specifications and their current status may be found on the FIPA Web site.

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

Contents

1 FIPA Contract Net Interaction Protocol 1
1.1 Explanation of the Protocol Flow 2
1.2 Exceptions to Interaction Protocol Flow 3
2 References 5
3 Informative Annex A — ChangeLog 6
3.1 2002/11/01 - version G by TC X2S 6

1 FIPA Contract Net Interaction Protocol

The FIPA Contract Net Interaction Protocol (IP) is a minor modification of the original contract net IP pattern[1] in that it adds rejection and confirmation communicative acts. In the contract net IP, one agent (the Initiator) takes the role of manager which wishes to have some task performed by one or more other agents (the Participants) and further wishes to optimise a function that characterizes the task. This characteristic is commonly expressed as the price, in some domain specific way, but could also be soonest time to completion, fair distribution of tasks, etc. For a given task, any number of the Participants may respond with a proposal; the rest must refuse. Negotiations then continue with the Participants that proposed.

The representation of this IP is given in *Figure 1* which is based on extensions to UML1.x. [Odell2001]. This protocol is identified by the token `fipa-contract-net` as the value of the `protocol` parameter of the ACL message.

[1] Originally developed by Smith and Davis.

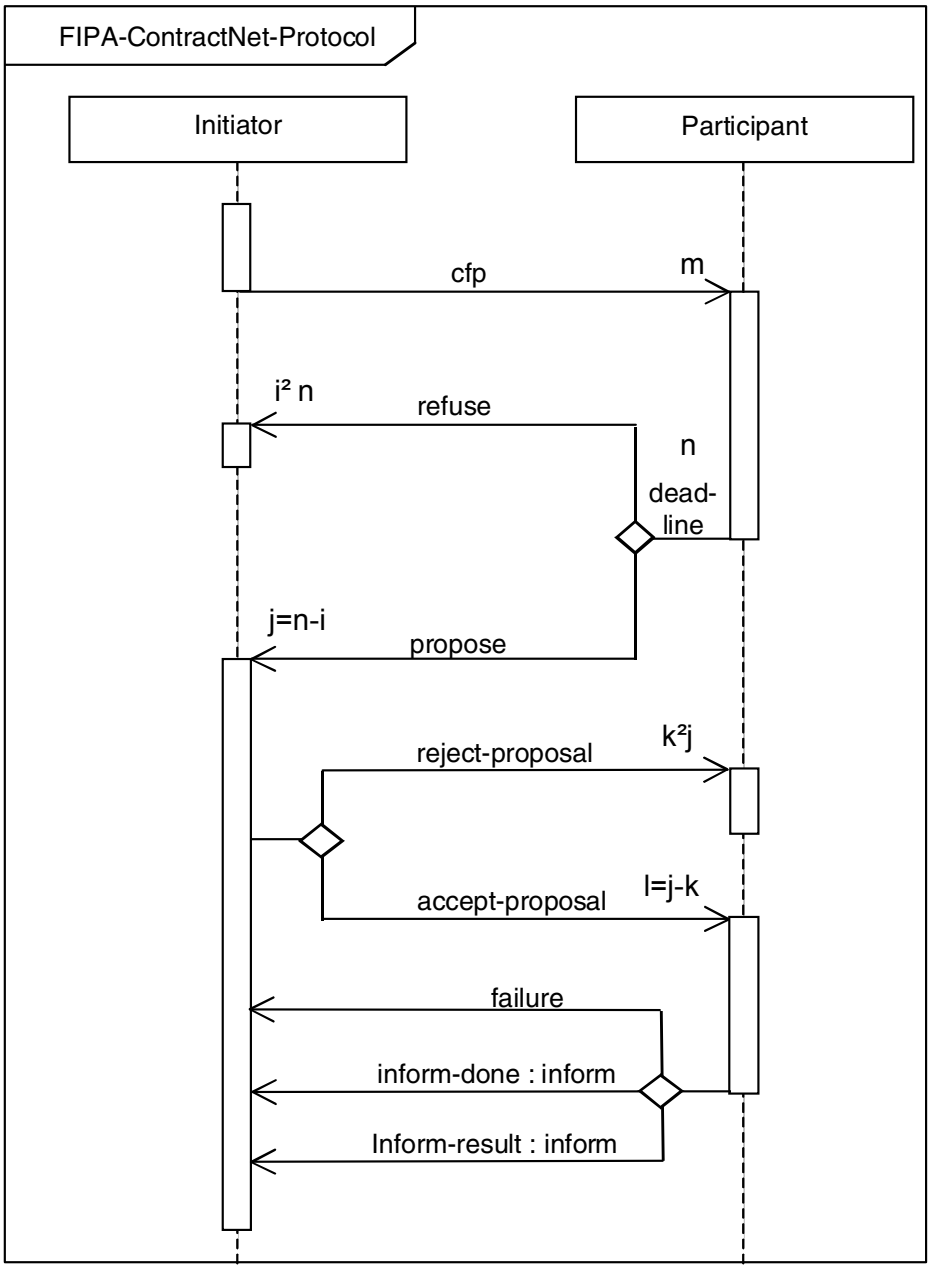

Figure 1: FIPA Contract Net Interaction Protocol

1.1 Explanation of the Protocol Flow

The Initiator solicits *m* proposals from other agents by issuing a call for proposals (`cfp`) act (see [FIPA00037]), which specifies the task, as well any conditions the Initiator is placing upon the execution of the task. Participants receiving the call for proposals are viewed as potential contractors and are able to generate *n* responses. Of these, *j* are proposals to perform the task, specified as `propose` acts (see [FIPA00037]).

The Participant's proposal includes the preconditions that the Participant is setting out for the task, which may be the price, time when the task will be done, etc. Alternatively, the *i=n-j* Participants may `refuse` (see [FIPA00037]) to propose. Once the deadline passes, the Initiator evaluates the received *j* proposals and selects agents to perform the task; one, several or no agents may be chosen. The *l* agents of the selected proposal(s) will be sent an `accept-`

`proposal` act (see [FIPA00037]) and the remaining *k* agents will receive a `reject-proposal` act (see [FIPA00037]). The proposals are binding on the Participant, so that once the Initiator accepts the proposal, the Participant acquires a commitment to perform the task. Once the Participant has completed the task, it sends a completion message to the Initiator in the form of an `inform-done` or a more explanatory version in the form of an `inform-result`. However, if the Participant fails to complete the task, a `failure` message is sent.

Note that this IP requires the Initiator to know when it has received all replies. In the case that a Participant fails to reply with either a `propose` or a `refuse` act, the Initiator may potentially be left waiting indefinitely. To guard against this, the `cfp` act includes a deadline by which replies should be received by the Initiator. Proposals received after the deadline are automatically rejected with the given reason that the proposal was late. The deadline is specified by the `reply-by` parameter in the ACL message.

Any interaction using this interaction protocol is identified by a globally unique, non-null `conversation-id` parameter, assigned by the Initiator. The agents involved in the interaction must tag all of its ACL messages with this conversation identifier. This enables each agent to manage its communication strategies and activities, for example, it allows an agent to identify individual conversations and to reason across historical records of conversations.

In the case of 1:N interaction protocols or sub-protocols the Initiator is free to decide if the same `conversation-id` parameter should be used or a new one should be issued. Additionally, the messages may specify other interaction-related information such as a timeout in the `reply-by` parameter that denotes the latest time by which the sending agent would like to have received the next message in the protocol flow.

1.2 Exceptions to Interaction Protocol Flow

At *any* point in the IP, the receiver of a communication can inform the sender that it did not understand what was communicated. This is accomplished by returning a `not-understood` message. As such, *Figure 1* does not depict a `not-understood` communication as it can occur at any point in the IP. The communication of a `not-understood` within an interaction protocol may terminate the entire IP and termination of the interaction may imply that any commitments made during the interaction are null and void. However, since this IP broadcasts to more than one Participant, multiple responses are also possible. Each response, then, must be evaluated separately – and some of these responses might be `not-understood`. However, terminating the entire IP in this case might not be appropriate, as other Participants may be continuing with their sub-protocols.

At any point in the IP, the initiator of the IP may cancel the interaction protocol by initiating the meta-protocol shown in *Figure 2*. The `conversation-id` parameter of the cancel interaction is identical to the `conversation-id` parameter of the interaction that the Initiator intends to cancel. The semantics of cancel should roughly be interpreted as meaning that the initiator is no longer interested in continuing the interaction and that it should be terminated in a manner acceptable to both the Initiator and the Participant. The Participant either informs the Initiator that the interaction is done using an `inform-done` or indicates the failure of the cancellation using a `failure`.

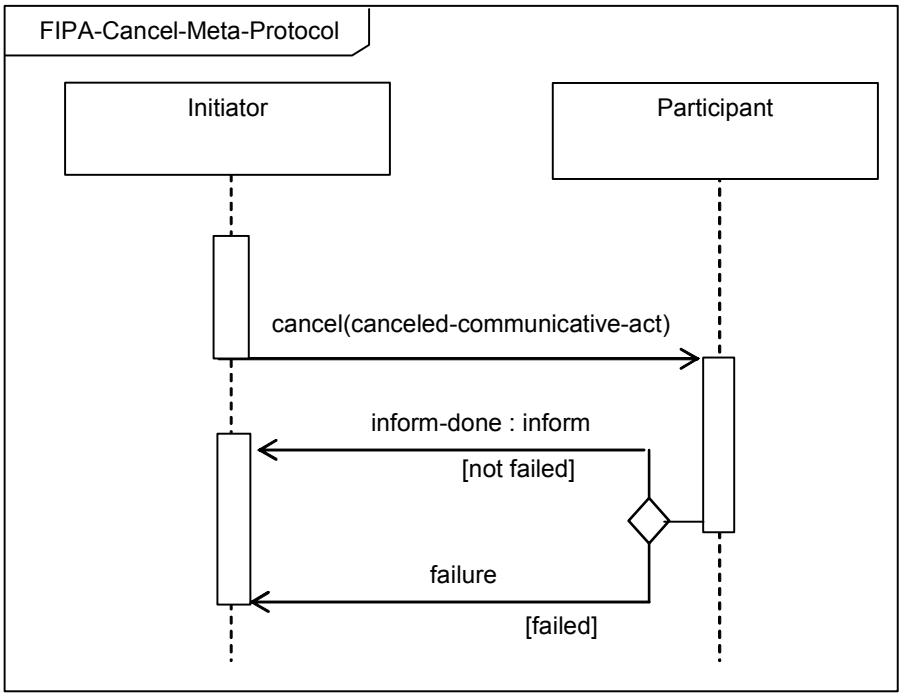

Figure 2: FIPA Cancel Meta-Protocol

This IP is a pattern for a simple interaction type. Elaboration on this pattern will almost certainly be necessary in order to specify all cases that might occur in an actual agent interaction. Real world issues such as the effects of cancelling actions, asynchrony, abnormal or unexpected IP termination, nested IPs, and the like, are explicitly not addressed here.

2 References

[FIPA00037] FIPA Communicative Act Library Specification. Foundation for Intelligent Physical Agents, 2000. http://www.fipa.org/specs/fipa00037/

[Odell2001] Odell, James, Van Dyke Parunak, H. and Bauer, B., *Representing Agent Interaction Protocols in UML*. In: Agent-Oriented Software Engineering, Ciancarini, P. and Wooldridge, M., Eds., Springer, pp. 121-140, Berlin, 2001. http://www.fipa.org/docs/input/f-in-00077/

3 Informative Annex A — ChangeLog

3.1 2002/11/01 - version G by TC X2S

Page 1, line 42: Reworked and expanded the section description of the IP

Page 2, Figure 1: The communication labeled `inform-ref` was changed to `inform-result` for clarity; the purpose of this communication is to inform the initiator of a result and `inform-result` implies `inform-done`

Page 2, Figure 1: The `not-understood` communication was removed

Page 2, Figure 1: To conform to UML 2, the protocol name was placed in a boundary, x is removed from the diamonds (xor is now the default) and the template box was removed

Page 2, line 72: Added a new section on Explanation of Protocol Flow

Page 2, line 72: Reworked and expanded the section on Exceptions of Protocol Flow to incorporate a meta-protocol for cancel

Page 2, line 72: Added a paragraph explaining the `not-understood` communication and its relationship with the IP