Tecnologia para Sistemas Inteligentes
Apontamentos para as aulas sobre

Sistemas de Agentes: Plataforma, linguagem de comunicação e protocolos de interação

Luís Miguel Botelho

Departamento de Ciências e Tecnologias da Informação
Instituto Superior de Ciências do Trabalho e da Empresa

Abril de 2012

Tecnologias para Sistemas Inteligentes

Apontamentos para as aulas

Índice

Sistemas de Agentes: Plataforma, linguagem de comunicação e protocolos de interação

Este documento apresenta o conceito de plataforma de agentes como um sistema de software que disponibiliza um conjunto de serviços que facilitam a existência e funcionamento de sociedades de agentes. O conceito é ilustrado através da descrição da Plataforma FIPA, uma plataforma de agentes standard, especificada pela FIPA.

Em seguida, o documento apresenta uma visão panorâmica da linguagem de comunicação FIPA ACL (Agent Commnication Language). A estrutura das mensagens FIPA ACL é apresentada através de um exemplo e da descrição de todos os seus parâmetros. Em seguida são apresentadas os vários grupos de mensagens (ação, informação, interrogação, negociação, delegação e erro). O conteúdo de cada mensagem é apresentado e explicado.

Finalmente, são apresentados alguns dos protocolos de interacção especificados pela FIPA: fipa-request, fipa-request-when, fipa-query, fipa-subscribe, fipa-contract-net, fipa-iterated-contract-net e ainda o meta protocolo fipa-cancel.

1 Plataforma FIPA

O capítulo introdutório relativo ao assunto dos sistemas de agentes e suas arquitecturas incidiu essencialmente sobre arquitecturas de sistemas multi-agente. Neste contexto, arquitectura refere-se ao modo como os agentes se organizam no sistema e á forma como o controlo e a comunicação são implementados. Neste capítulo, apresenta-se a noção relacionada de plataforma. Uma plataforma para agentes é uma infraestrutura computacional que pode prestar um conjunto de serviços, os quais permitem a implementação de um ou mais tipos de arquitecturas de sistemas de agentes (idealmente, todos os tipos de arquitecturas). De entre os serviços existentes numa plataforma destacam-se o serviço de gestão da plataforma, o serviço de páginas amarelas, o serviço de páginas brancas, o serviço de transporte e encaminhamento de mensagens. A plataforma FIPA inclui todos esses serviços.

A FIPA ("Foundation for Intelligent Physical Agents") é uma organização internacional cujos sócios são essencialmente grandes empresas ligadas à indústria dos computadores e seus principais clientes (IBM, SUN Microsystems, British Telecom e a Boieng, entre outras) e cujo principal objectivo é a definição de normas ("standards") para tecnologias conceptuais e computacionais usadas para a criação e operação de agentes e de sistemas multi-agentes.

De acordo com as suas mais recentes especificações (2002), foi definida uma plataforma FIPA concreta em que os serviços de páginas brancas e de gestão da plataforma estão a cargo de um agente chamado AMS (*"Agent Management System"*); o serviço de páginas amarelas está a cargo de um agente chamado DF (*"Directory Facilitator"*); e em que o serviço de transporte de mensagens é prestado pela própria plataforma mas não corresponde a nenhum agente. Nas especificações de 1997, o serviço de transporte de mensagens era também prestado por um agente chamado ACC (*"Agent Communication Channel"*).

De acordo com as especificações da FIPA, o AMS mantém uma lista dos nomes e endereços de todos os agentes da plataforma; o DF mantém uma lista com as descrições de serviços prestados por agentes da plataforma (e eventualmente de outras plataformas) juntamente com os nomes dos agentes que os prestam; o serviço de transporte de mensagens (MTS – *"Message Transport Service"*) encarrega-se de distribuir mensagens recebidas na plataforma para qualquer dos seus agentes e de contactar o MTS de outra plataforma para garantir a distribuição de mensagens enviadas por um dos agentes da plataforma para agentes de outra plataforma. Uma plataforma FIPA tem um único AMS e um único MTS, mas pode ter diversos DFs.

No restante destes apontamentos, será adoptada a perspectiva mais recente em que existem os agentes AMS e DF, mas não o agente ACC.

Além dos serviços já referidos, uma plataforma FIPA poderá prestar outros serviços (por exemplo, o serviço de ontologia) os quais poderão estar a cargo de agentes ou da própria plataforma. De acordo com as especificações FIPA, alguns serviços podem não existir, mas se existirem têm que concordar com as especificações definidas pela FIPA.

Um sistema multi-agente desenvolvido sobre uma plataforma FIPA será constituído por todos os agentes da plataforma e ainda pelos agentes específicos da aplicação. Um sistema multi-agente pode estender-se por uma ou mais plataformas. E uma plataforma pode estender-se por um ou mais computadores.

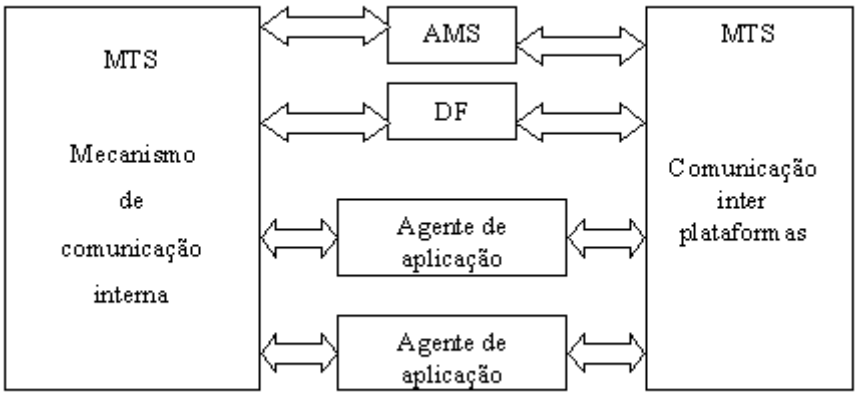

Figura 1 – Plataforma FIPA com dois agentes de aplicação

A Figura 1 representa uma plataforma FIPA com dois agentes de aplicação, um DF e o AMS, os quais comunicam entre si através do MTS.

Numa situação típica, em especial se for usada uma arquitectura de negociação, quando um agente precisa que uma dada tarefa seja executada por outro agente pede ao DF para o informar dos nomes dos agentes capazes de a realizar. Depois poderá pedir aos agentes indicados pelo DF para apresentarem as suas propostas para a realização da tarefa. Alguns destes agentes apresentam as suas propostas, as quais são avaliadas e sujeitas à selecção do agente que as solicitou. O agente que solicitou as propostas envia mensagens de rejeição de proposta a alguns dos agentes e mensagens de aceitação de proposta aos outros. Os agentes que recebem a mensagem de aceitação de proposta ficam obrigados a prestar o serviço proposto nas condições da proposta.

O mecanismo de coordenação descrito é uma das possibilidades de vários agentes coordenarem a sua acção na satisfação dos seus objectivos. Este mecanismo de coordenação assenta na utilização de mensagens definidas na linguagem de comunicação de agentes da FIPA. Esta linguagem chama-se ACL ("Agent communication language"). Existem outras possibilidades de coordenação de agentes que usam outras mensagens da linguagem ACL. A secções 2 descreve a linguagem FIPA ACL em maior detalhe. Os protocolos de interacção definidos pela FIPA são apresentados na secção 3. Outros aspectos da interacção de agentes, cuja explicação é mais extensa, são apresentados em componentes distintos da documentação.

2 Linguagem de Comunicação FIPA-ACL

A especificação [FIPA 2002-37] define a linguagem ACL de comunicação entre agentes. Para a FIPA, um agente é um programa ou uma máquina que fala ACL.

Além da linguagem de comunicação, as mensagens têm conteúdo, o qual pode ser escrito em qualquer linguagem capaz de representar proposições, acções e expressões de identificação ("Identifying Referential Expressions").

Todas as mensagens ACL têm a especificação do tipo da mensagem e um conjunto de parâmetros dos quais só muito poucos são obrigatórios. A Figura 2 mostra o aspecto de uma mensagem do tipo *inform* com alguns parâmetros.

```
(inform
   :sender a
   :receiver (set b)
   :content "((humidade-relativa 80))"
   :language fipa-sl
   :ontology (set metereologia)
)
```

Figura 2 – Mensagem ACL com alguns parâmetros

As mensagens ACL têm todas um aspecto semelhante ao da Figura 2. O tipo da mensagem é ***inform***. Todos os parâmetros são indicados pelo nome do parâmetro precedido por dois pontos (:). Os parâmetros *:sender* e *:receiver* especificam o remetente e o destinatário da mensagem, respectivamente. Como uma mensagem pode ser enviada a mais do que um agente, o valor do parâmetro *:receiver* é um conjunto, daí o operador *set*. Os parametros :*content*, *:language* e *:ontology* referem-se todos ao conteúdo da mensagem, respectivamente o conteúdo propriamente dito, a linguagem em que o conteúdo é escrito e as suas ontologias (i.e., os conceitos do domínio da aplicação).

A tabela enumera todos os parâmetros pre-definidos das mensagens ACL. A maioria desses parâmetros não são obrigatórios. A maioria das mensagens não usa todos eles.

Parâmetro	Tipo	Descrição
`performative`	Enumerado (qualquer das performativas)	Tipo de acto comunicativo (e.g., *inform* e *request*)
`sender`	agent-identifier	Agente que envia a mensagem
`receiver`	set of agent-identifier	Conjunto dos agentes a quem a mensagem é enviada. É o único parâmetro obrigatório em todas as mensagens.
`reply-to`	agent-identifier	Indica a quem é que a mesagem subsequente a esta na conversação deve ser enviada
`content`	String	Conteúdo da mensagem, expresso numa dada linguagem de conteúdo. O conteúdo só pode ser interpretado sabendo o tipo do acto comunicativo
`language`	Word	Linguagem em que o conteúdo é representado
`encoding`	Word	Codificação do conteúdo. Se não for especificado, assume-se que a codificação é a indicada no envelope da mensagem
`ontology`	Set of String	Conjunto de ontologias do conteúdo
`protocol`	Word	Protocolo de interação (e.g., fipa-request)
`conversation-id`	String	Identificador único criado pelo iniciador da conversa que identifica a conversação a que a mensagem pertence
`reply-with`	Word	Identificador da mensagem. É usado para que a resposta possa identificar a mensagem que lhe deu origem
`in-reply-to`	Word	Identificador da mensagem a que a mensagem actual responde
`reply-by`	Datetime	Especifica quando é que é esperada a resposta

Tabela 1 - Parâmetros das mensagens ACL

Além dos parâmetros especificados na Tabela 1, as mensagens ACL podem ter parâmetros adicionais, específicos da aplicação, os quais são da responsabilidade da equipa de desenvolvimento. Esses parâmetros devem começar com a letra x.

Existem diversos grupos de mensagens ACL. Mensagens para informar, mensagens para requerer a execução de acções, mensagens para interrogar, mensagens para negociar, mensagens para delegar, e mensagens de erros. O tipo da mensagem especifica inequivocamente o tipo do seu conteúdo. A Tabela 2 apresenta a relação entre cada tipo de mensagem e o seu tipo de conteúdo.

Tipo	Grupo	Conteúdo
accept-proposal	negociação	acção, condição (proposição)
agree	acção	acção (termo) e condição de aceitação
cancel	acção	acção
Cfp	negociação	proposta: (acção, condição)
confirm	informação	afirmação: proposição
disconfirm	informação	afirmação: proposição
failure	acção	acção, razão (proposição)
inform	informação	afirmação: proposição
not-understood	processamento de erros	mensagem, razão
propagate	delegação	Expressão referencial descrevendo agentes, uma mensagem, e uma condição
propose	negociação	proposta (acção, condição)
proxy	delegação	Expressão referencial descrevendo agentes, uma mensagem, e uma condição
query-if	interrogação	proposição
query-ref	interrogação	expressão de identificação
refuse	acção	acção, razão
reject-proposal	negociação	proposta (i.e., acção, condição), razão
request	acção	acção
request-when	acção	acção, condição
request-whenever	acção	acção, condição
subscribe	interrogação	expressão de identificação

Tabela 2 - Os vários tipos de mensagens ACL e respectivos conteúdos

A Tabela 3 descreve sumária e informalmente cada tipo de mensagem ACL.

Tipo	Descrição sumária
accept-proposal	Mensagem usada por um agente para aceitar a proposta apresentada por outro agente, mediante determinadas condições (propose).
agree	Mensagem usada por um agente para informar um segundo agente que, nas condições especificadas na mensagem, aceita executar a acção requerida pelo segundo (request)
cancel	Mensagem usada para cancelar um pedido efectuado anteriormente (request)
Cfp	Mensagem para pedir a outros agentes que apresentem as suas propostas para a execução de uma dada tarefa
confirm	Mensagem usada para confirmar uma crença (mais especificamente, uma incerteza – operador U)
disconfirm	Mensagem usada para desconfirmar uma crença
failure	Mensagem usada por um agente para informar um segundo agente que a tentativa de execução da acção requerida (request) ou proposta (propose) falhou.
inform	Mensagem para informar um agente que uma dada proposição é verdadeira.
not-understood	Mensagem para informar um agente de que uma dada mensagem não foi percebida por uma dada razão
propagate	Mensagem usada para enviar ao receptor uma mensagem embebida no *propagate* e para lhe pedir para propagar a mensagem *propagate* ao conjunto de agentes cujos identificadores resultam da avaliação da expressão de referência especificada. O *propagate* e a mensagem embebida deverão ser propagadas quando o agente que recebe a mensagem *propagate* achar que a condição especificada se verifica.
propose	Mensagem em que um agente se propõe executar uma dada acção numa dada condição
proxy	Mensagem usada para pedir ao receptor que envie a mensagem embebida no *proxy* ao conjunto de agentes cujos identificadores resultam da avaliação da expressão de referência especificada. A mensagem embebida deverá ser enviada quando o agente que recebe a mensagem *proxy* achar que a condição especificada se verifica.
query-if	Mensagem em que o emissor pergunta se uma dada proposição é verdadeira
query-ref	Mensagem em que o emissor pede ao receptor para lhe dizer quais os objectos que satisfazem uma dada proposição.
refuse	Mensagem em que o emissor recusa executar uma acção pedida pelo receptor
reject-proposal	Mensagem em que o emissor rejeita a proposta efectuada pelo receptor, por uma dada razão
request	Mensagem em que o emissor pede ao receptor que execute uma dada acção
request-when	Mensagem em que o emissor pede ao receptor que execute uma dada acção quando uma dada condição se verificar.
request-whenever	Mensagem em que o emissor pede ao receptor que execute uma dada acção sempre que uma dada condição se verificar.
Subscribe	Mensagem em que o emissor pede ao receptor para lhe enviar os objectos que satisfazem uma dada condição, sempre que existirem objectos que a satisfaçam.

Tabela 3 - Breve descrição de mensagens FIPA

O conjunto de mensagens descritas nos parágrafos e figuras precedentes é suficientemente lato para se poder criar um grande leque de aplicações de agentes nos mais diversos domínios.

Além de definir os tipos e categorias das mensagens, os seus parâmetros e os conteúdos, A FIPA define também as pre-condições para que as mensagens possam ser enviadas (FP, "feasibility preconditions"), e os efeitos que é lícito esperar quando as mensagens são recebidas pelo seus destinatários (RE, *"Rational Effects"*) [FIPA 2002-37]. Apesar de muito útil, esta especificação dos FPs e REs de cada mensagem cai fora do âmbito deste texto.

3 Protocolos de Interacção

Os protocolos de interacção servem para especificar o comportamento esperado dos agentes durante a sua interacção com outros agentes. Por exemplo, um protcolo de interacção pode especificar que, quando um agente recebe uma mensagem Request com um dado pedido, só pode responder com a mensagem Not-Understood (se não percebe algum aspecto da mensagem recebida), ou com a mensagem Refuse (se não aceita o pedido que lhe é feito), ou com a mensagem Agree (se aceita o pedido que lhe é feito).

A utilização de protocolos de interacção não é obrigatória mas é uma boa ajuda para o desenho de aplicações baseadas em agentes porque permite reduzir o número de alternativas que um agente tem que considerar em cada passo da sua interacção com outros.

Têm sido propostos diversos formalismos, quase sempre de natureza gráfica, para especificar protocolos de interacção. Alguns dos mais conhecidos são os Diagramas de Protocolo da linguagem AUML (Agent Unified Modelling Language) propostos por [Odell et al. 2000] [Bauer et al. 2001]. Os Diagramas de Protocolo AUML são uma extensão aos diagramas de sequência UML [Booch et al. 1999]. As Redes de Petri Coloridas (*colored Petri nets*) [Nowostawski et al. 2001] são também muito usados na especificação de protocolos. As redes de Petri coloridas e o AUML têm capacidades semelhantes. Também os diagramas de transição de estados foram propostos para representar protocolos de interacção mas, tendo menor capacidade de expressão que os outros dois mecanismos referidos, têm perdido terreno face à concorrência.

Neste texto usaremos os Diagramas de Protocolo, os quais têm sido utilizados nas especificações de protocolos de interacção da FIPA. Este tipo de especificação é mais útil para quem desenha e implementa agentes do que para os agentes em si, pois é uma especificação com pouco detalhe que não serve para a geração automática de código. No entanto, o AUML tem outros tipos de diagrama que podem ser usados em combinação com os Diagramas de Protocolo. Usando todo o poder do AUML é possível a geração automática de código para a execução de protocolos de interacção.

Os Diagramas de Protocolo AUML definem um protocolo indicando o seu nome e especificando o comportamento de cada um dos intervenientes na conversação. O comportamento dos intervenientes é condicionado pela especificação de vários papéis (*agent roles*) do protocolo. Muitos dos protocolos de interacção têm apenas dois papéis: o iniciador do protocolo (aquele agente que inicia o protocolo através do envio de uma dado tipo de mensagem) e o participante (aquele a quem o iniciador envia a mensagem inicial). No entanto, há protocolos com mais que dois papéis.

Certos papéis de certos protocolos podem ser desempenhados em simultâneo por mais que um único agente, enquanto outros papéis só podem ser desempenhados por um único agente (em cada conversação). Finalmente, o mesmo agente pode desempenhar mais que um papel num protocolo.

Na explicações e exemplos apresentados usa-se a linguagem de comunicação FIPA ACL [FIPA 2002-37] e a linguagem de conteúdo FIPA SL [FIPA 2002-8].

Nas próximas secções apresentaremos os protocolos de interacção definidos pela FIPA.

3.1 Protocolos de Interacção da FIPA

A FIPA definiu uma grande diversidade de protocolos de interacção que podem ser usados em muitas situações de interacção entre agentes, nomeadamente em interacções em que um agente pede a outro para executar uma dada acção, em situações de pergunta / resposta, em negociação entre agentes, e para delegação de tarefas.

Esta secção faz uma apresentação sumária dos protocolos definidos pela FIPA e do estado da especificação correspondente (standard, experimental, preliminar). As próximas secções descrevem em maior detalhe alguns dos protocolos definidos. Em qualquer caso, um protocolo pode terminar abruptamente em duas situações:

1. O agente que recebe uma mensagem não compreende a mensagem recebida, caso em que tem de enviar uma mensagem *not-understood* cujo conteúdo é a mensagem não compreendida e a razão pela qual ela não foi compreendida; e
2. O agente iniciador do protocolo resolve cancelar o pedido efectuado. Neste caso, deve ser enviada uma mensagem *cancel* cujo conteúdo é a mensagem em que o pedido foi feito. A mensagem *cancel* pode iniciar uma conversação governada pelo protocolo FIPA-Cancel, o qual se encontra descrito na secção 3.6.

Nenhuma destas situações excepcionais será representada nos diagramas que especificam os protocolos de interacção apresentados e explicados nas secções seguintes.

Notas sobre a linguagem de representação dos protocolos

A linguagem gráfica usada para especificar protocolos de interacção (Diagramas de Protocolos AUML) é essencialmente a linguagem dos Diagramas de Sequências da UML com algumas extensões. A Figura 3 ilustra uma das extensões feitas aos Diagramas de Sequências UML.

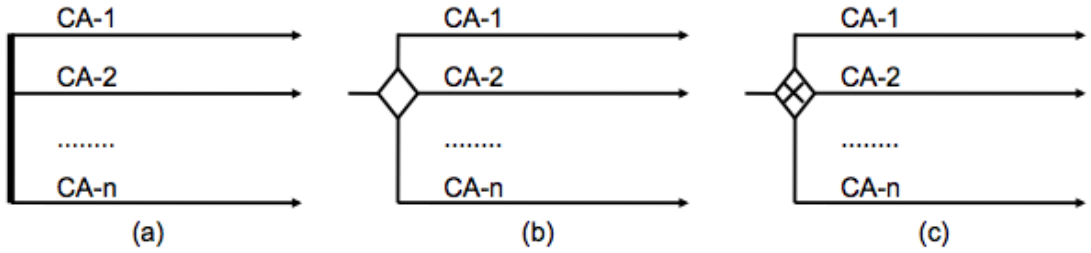

Figura 3 – Especificação de Protocolos de Intercção

Todas as situações representadas na figura representam *threads* múltiplas. Na figura (a), todas as mensagens CA-1, CA-2, ..., CA-n são enviadas em paralelo. Em (b), zero ou mais mesnsagens são enviadas. Se mais do que uma mensagem for enviada, elas serão enviadas em paralelo. Finalmente, em (c), exactamente uma mensagem é enviada.

Protocolos de pergunta e resposta

Talvez a maior parte das interacções de agentes de informação seja a apresentação de perguntas e a recepção das respectivas respostas. A FIPA definiu dois protocolos de interacção que podem ser usados nestas circunstâncias: FIPA-Query [FIPA 2002-27] e FIPA-Subscribe [FIPA 2002-35].

O FIPA-Query é usado quando um agente pretende fazer uma pergunta e receber uma resposta apenas. O protocolo permite regular as interacções em que a pergunta é fechada[1], e as interacções em que a pergunta é aberta. O protocolo FIPA-Query é iniciado por uma mensagem *query-ref* ou por uma mensagem *query-if*. A mensagem *query-ref* é usada para perguntas abertas. A mensagem *query-if* é usada para perguntas fechadas.

O FIPA-Subscribe é usado quando um agente pretende subscrever um determinado tipo de informação. Quando o receptor da mensagem *subscribe* acredita que tem informação relativa à pergunta recebida, envia uma mensagem informativa ao iniciador do protocolo com essa informação. O protocolo FIPA-Subscribe é iniciado pela mensagem *subscribe*.

Tanto a especificação do FIPA-Query como a especificação do FIPA-Subscribe são especificações standard.

Protocolos de Pedido de Execução de Acções

Existem dois protocolos usados pelos agentes nas suas interacções para pedirem a outro a execução de uma ou mais acções: FIPA-Request [FIPA 2002-26] e FIPA-Request-When [FIPA 2002-28].

[1] Uma pergunta fechada é uma pergunta cuja resposta só pode ser do tipo sim/não (ou True/false)

O FIPA-Request é usado para pedir a execução incondicional de uma ou mais acções. O FIPA-Request-When é usado para pedir a execução da acção ou das acções especificadas na mensagem que inicia o protocolo apenas quando uma dada condição se verificar.

O FIPA-Request é iniciado pela mensagem *request*. O FIPA-Request-When é iniciado pela mensagem *request-when*.

As especificações dos dois protocolos são especificações standard.

Protocolos de negociação entre agentes

Existem cinco protocolos de negociação entre agentes: FIPA-Propose [FIPA 2002-36], FIPA-Contract-Net [FIPA 2002-29], FIPA-Iteratedt-Contract-Net [FIPA 2002-30], FIPA-English-Auction [FIPA 2001-31], e FIPA-Dutch-Auction [FIPA 2001-32].

O mais simples de todos, o FIPA-Propose usa-se quando um agente se propõe realizar uma dada acção em determinadas condições. Este protocolo é iniciado pela mensagem *propose*. Deve notar-se que nem sempre que a mensagem *propose* é usada se inicia uma ocorrência do protocolo FIPA-Propose. Por um lado, poderá haver outros protocolos alternativos que podem ser usados com o mesmo propósito. Por outro lado, a mensagem *propose* poderá ser parte de conversações reguladas por outros protocolos mais complexos, como é o caso do FIPA-Contract-Net.

O FIPA-Contract-Net é usado quando um agente pede a um ou mais agentes para lhe apresentarem propostas para a execução de uma dada acção (ou acções). Este protocolo é iniciado pelo envio da mensagem *cfp* (*call for proposals*) e procede, salvo nos casos em que a mensagem não é compreendida, pela apresentação de propostas ou pela recusa de participar no protocolo.

O FIPA-Iterated-Contract-Net é uma versão iterada do FIPA-Contract-Net em que o iniciador, após ter recebido propostas dos outros intervenientes, decide voltar a pedir a apresentação de novas propostas porque não ficou satisfeito com as propostas recebidas. Tal como o FIPA-Contract-Net, o FIPA-Iterated-Contract-Net é iniciado pelo envio da mensagem *cfp*. Os agentes intervenientes na conversação distinguem a os dois protocolos através do valor do parâmetro *:protocol* da mensagem *cfp* que inicia a conversação.

Os protocolos FIPA-English-Auction e FIPA-Dutch-Auction são usados em situações de negociação com a configuração de um leilão. No caso do FIPA-English-Auction, o leilão é do tipo inglês ou ascendente (cada licitação tem que ser superior ao valor da última licitação). No caso do FIPA-Dutch-Auction, o leilão é de tipo holandês ou descendente (o preço vai descendo e a primeira licitação determina o preço da venda). Tanto o leilão inglês como o leilão holandês são iniciados por uma mensagem *inform* em que se informa os outos intervenientes que o leilão começou.

As especificações dos protocolos FIPA-Propose, FIPA-Contract-Net, e FIPA-Iteratedt-Contract-Net são standards. As especificações dos protocolos FIPA-English-Auction e FIPA-Dutch-Auction são experimentais.

Protocolos para delegação de pedidos

Os protcolos FIPA-Brokering [FIPA 2002-33] e FIPA-Recruiting [FIPA 2002-34] são protocolos de delegação. No caso do FIPA-Brokering, o agente iniciador pede ao agente intermediário (*broker*) para este encontrar um ou mais agentes que possam responder a uma dada pergunta. O intermediário encontra os agentes capazes de responder, passa-lhes a pergunta, recebe a resposta e envia a resposta para o agente iniciador. O FIPA-Brokering é iniciado pelo envio de uma mensagem *proxy*.

O protocolo FIPA-Recruiting é um protocolo muito semelhante ao protocolo de brokering acabado de descrever. A diferença é que a resposta dos agentes contactados pelo agente intermediário vai directamente para o agente iniciador do protocolo ou para um conjunto especificado de outros agentes. O protocolo FIPA-Recruiting é iniciado pelo envio de uma mensagem *proxy*. O protocolo FIPA-Recruiting é o único da família de protocolos definidos pela FIPA que tem mais que dois papéis: iniciador, recrutador e receptor designado.

3.2 *Protocolo FIPA-Request*

O protocolo FIPA-Request é iniciado por um agente que envia a mensagem *request* a outro agente para lhe requerer a execução de uma ou mais acções (Figura 4).

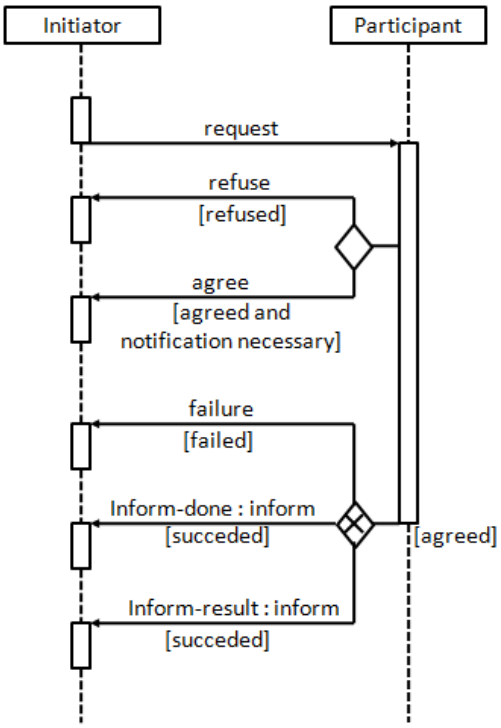

Figura 4 – Protocolo FIPA-Request

Como se pode ver na Figura 4, o protocolo FIPA-Request tem dois papéis: o iniciador, que é o agente que inicia o protocolo enviando o pedido; e participante, que é o agente que recebe o pedido e que, se concordar, o deverá executar.

Quando o agente partcipante recebe a mensagem *request* enviada pelo iniciador, tem duas alternativas: ou responde com a mensagem *refuse* indicando que rejeita o pedido apresentado e a razão da recusa; ou envia a mensagem *agree* indicando que aceita o pedido apresentado.

Se o participante aceita o pedido apresentado, fica comprometido a garantir que o pedido seja executado, e o protocolo passa para o estado *agreed* em que se espera um determinado tipo de comportamento. Se o participante recusa o pedido apresentado, o protocolo termina, isto é, passa para o seu estado final.

Se a execução do protocolo se encontra no estado *agreed*, espera-se que o pedido aceite seja executado. Três coisas podem acontecer: a execução da acção pedida falha, caso em que o participante tem que enviar a mensagem *failure* ao iniciador dizendo que a execução da acção pedida falhou e qual a razão dessa falha; ou envia uma mensagem *inform* ao iniciador dizendo que a acção pedida foi executada com sucesso; ou envia uma mensagem *inform* ao iniciador com o resultado da execução da acção pedida. Em qualquer um destes casos, a execução do protocolo termina, isto é, o protocolo transita para o seu estado final.

Quando a execução da acção pedida resulta em informação a enviar ao iniciador, a mensagem *inform* enviada conterá a informação resultante. Este é o caso de acções de procura de informação numa base de dados interna por exemplo. É o que acontece quando se pede ao agente DF (*Directory Facilitator*) para procurar agentes que prestem um determinado serviço.

Quando a execução da acção pedida não resulta em informação a ser enviada ao iniciador, a mensagem *inform* enviada dirá apenas que a acção foi executada com sucesso.

Todas as mensagens pertencentes à mesma conversação devem ter o mesmo valor para o parâmetro *:conversation-id*, o qual é estabelecido pelo iniciador do protocolo quando envia a mensagem *request* inicial.

O protocolo FIPA-Request está na base de outros protocolos como é o caso do FIPA-Query.

3.3 Protocolo FIPA-Request-When

O protocolo FIPA-Request-When é iniciado por um agente que envia a mensagem *request-when* a outro agente para lhe requerer a execução de uma ou mais acções, quando se verificar uma dada condição também especificada na mensagem (Figura 5).

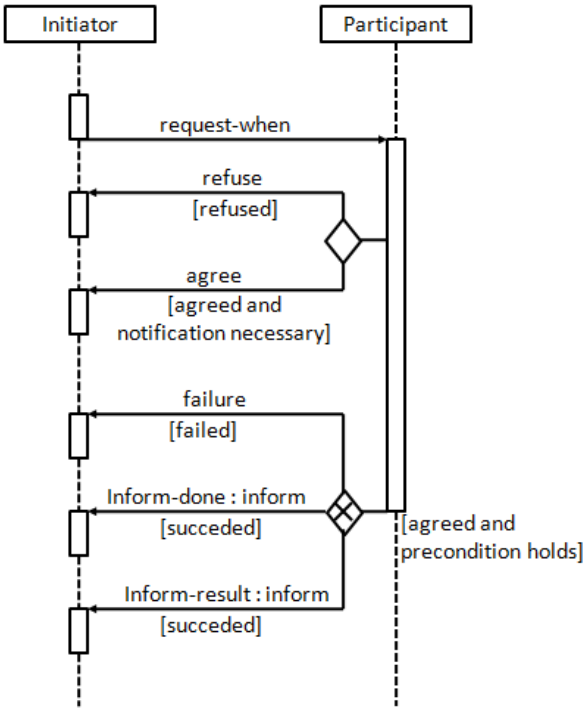

Figura 5 – Protocolo FIPA-Request-When

Quando o participante recebe a mensagem *request-when* enviada pelo iniciador para iniciar o protocolo, pode fazer duas coisas: enviar a mensagem *refuse*, indicando que rejeitou o pedido apresentado e indicando a razão da recusa; ou enviar a mensagem *agree* indicando a sua aceitação do pedido apresentado. Neste último caso, o protocolo transita para o estado *agreed* em que se espera um dado comportamento do agente participante.

Se o protocolo está no estado *agreed* e a condição especificada na mensagem *request-when* que inicia o protocolo estiver satisfeita, espera-se que a acção pedida seja executada. Três coisas podem acontecer: ou a execução da acção pedida falha caso em que o participante envia uma mensagem *failure* ao iniciador dizendo que a execução da acção falhou e qual a razão da falha; ou o partticipante envia uma mensagem *inform* contendo o resultado da execução da acção ou contendo a indicação de que a acção pedida foi executada com sucesso. Em qualquer destes casos, a conversação iniciada pela mensagem *request-when* termina.

Todas as mensagens pertencentes à mesma conversação devem ter o mesmo valor para o parâmetro *:conversation-id*, o qual é estabelecido pelo iniciador do protocolo quando envia a mensagem *request-when* inicial.

O protocolo FIPA-Request-When consiste apenas numa alteração do protocolo FIPA-Request, em que se especifica uma condição para a execução da acção.

3.4 Protocolo FIPA-Query

Uma pergunta é um caso particular de um pedido de informação. Há dois tipos de perguntas: perguntas abertas e perguntas fechadas. Uma pergunta fechada é um pedido em que o agente que envia a mensagem com a pergunta pede ao agente que a recebe para este lhe dizer se uma dada afirmação é

verdadeira ou falsa. Uma pergunta aberta é um pedido em que o agente que envia a mensagem com a pergunta pede ao agente que a recebe para este lhe enviar informação descrita através de uma dada condição. Em qualquer caso, uma pergunta é um caso particular de um pedido. Consequentemente, o protocolo definido pela FIPA que governa as interacções iniciadas com perguntas é muito semelhante ao protocolo que governa as interacções iniciadas por pedidos (*FIPA-Request-Protocol*).

O protocolo FIPA-Query foi definido pela FIPA para governar interacções iniciadas quer por perguntas abertas (iniciadas pela mensagem *query-ref*), quer por perguntas fechadas (iniciadas pela mensagem *query-if*). Nesse protocolo, o agente iniciador envia uma mensagem *query-if* ou uma mensagem *query-ref*. É esta mensagem que inicia o protocolo.

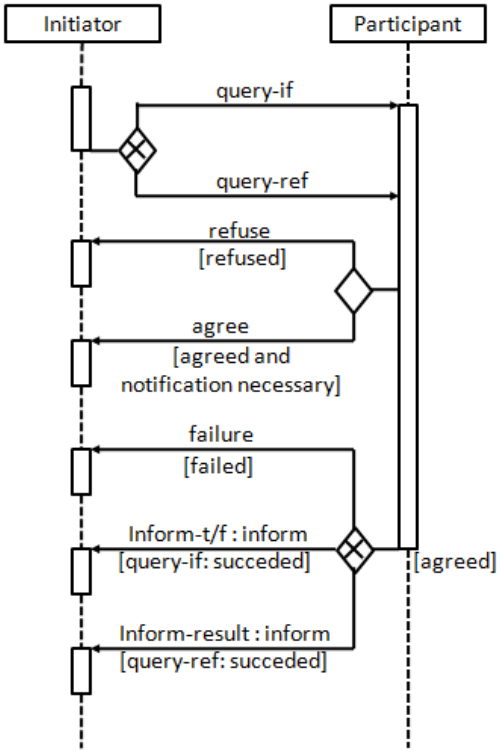

Figura 6 – FIPA-Query Protocol

Tal como acontecia no protocolo FIPA-Request, o agente que recebe a pergunta pode recusar responder-lhe (através da mensagem *refuse* com a razão da recusa). Se o agente não recusar responder, o protocolo transita para o estado *agreed*. Neste estado, se for necessário enviar uma notificação, o agente que recebeu a pergunta (*Participant*) envia uma mesnagem *agree* ao agente que a enviou (*Initiator*) dizendo que vai responder à pergunta.

No estado *agreed*, o agente *Participant* tem de tentar responder. Se não conseguir responder (por exemplo, porque a pergunta assumia que deveria existir exactamente uma resposta, mas o agente encontra mais de uma resposta), o *Participant* envia a mesnagem *failure* dizendo a causa da falha. Se a não tiver ocorrido uma falha, o agente tem de enviar a resposta ao *Initiator*. Se a pergunta recebida foi uma pergunta aberta, a resposta é uma mensagem *inform* com a informação que satisfaz a condição usada na pergunta (*inform-result*, no protocolo). Se a pergunta recebida é uma pergunta fechada, a resposta é uma mensagem *inform* com a proposição da pergunta (se o agente acreditar que esta é verdadeira), ou com a negação da proposição da pergunta (se o agente acreditar que esta é falsa). Este caso está indicado pela notação *inform-t/f* no protocolo.

3.5 Protocolo FIPA-Subscribe

O protocolo FIPA-Subscribe é iniciado por um agente que envia a mensagem *subscribe* a outro agente para lhe requerer o envio de dada informação sempre que o receptor acreditar que existe informação consistente com o pedido efectuado. A mensagem *subscribe* que inicia a conversação contém uma

expressão de referência que caracteriza a informação pretendida. Este protocolo contém uma repetição do envio da informação pedida (Figura 7).

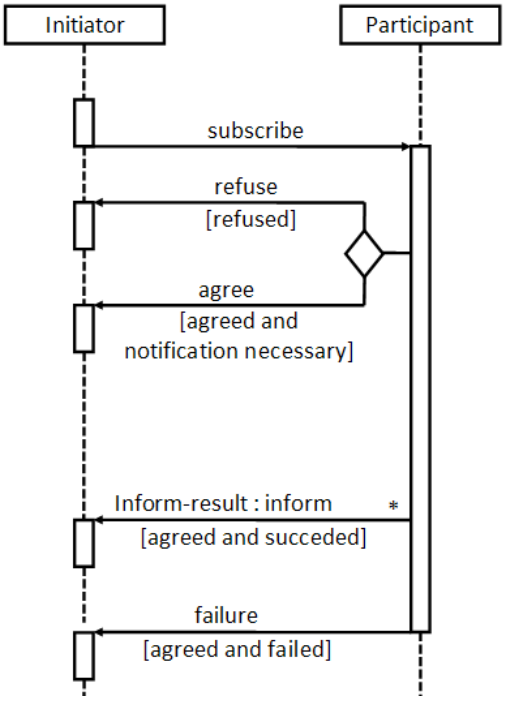

Figura 7 – Protocolo FIPA-Subscribe

Após a recepção da mensagem *subscribe*, o participante decide se aceita ou não enviar a informação pedida pelo iniciador. Se decide rejeitar o pedido de informação, envia a mensagem *refuse* com a razão da recusa, e o protoclo termina. Se decide aceitar o pedido de informação, o participante envia uma mensagem *agree*, e o protocolo transita para o estado *agreed*.

No estado *agreed*, o participante tem que enviar mensagens ao iniciador contendo a informação referida na mensagem recebida sempre que o participante acreditar que dispõe de infomação consistente com a especificação recebida na mensagem *subscribe* que iniciou a conversação. Se, durante este processo, ocorrer uma falha, o participante envia uma mensagem *failure* ao iniciador informando-o de que uma falha ocorreu e a razão de ser dessa falha. A mensagem *failure* termina o protocolo.

Todas as mensagens pertencentes à mesma conversação são identificadas por um identificador único enviado como valor do parâmetro *:conversation-id* da mensagem. Este identificador é gerado pelo agente iniciador quando envia a mensagem *subscribe* que inicia a conversação.

Se o participante aceitar participar no protocolo e se nenhuma falha ocorrer, a conversação não termina, a não ser que o iniciador cancele o pedido através da mensagem *cancel*. Esta eventualidade não faz parte do protocolo FIPA-Subscribe; em vez disso, existe um meta-protocolo para que especifica o cancelamento de qualquer conversação (secção 3.6).

3.6 Meta Protocolo FIPA-Cancel

O meta-protocolo FIPA-Cancel tem dois papéis: o iniciador que envia uma mensagem *cancel*, e o participante que recebe a mensagem. Este meta-protocolo é usado para cancelar conversações previamente iniciadas pelo agente que envia a mensagem *cancel* (Figura 8).

O FIPA-Cancel é iniciado quando o iniciador envia uma mensagem *cancel* cujo conteúdo é a mensagem que iniciou a conversação que se pretende cancelar. O valor do parâmetro *:conversation-id* da conversa iniciada com a mensagem *cancel* é idêntico ao valor do parâmetro *:conversation-id* da conversa que se pretende cancelar.

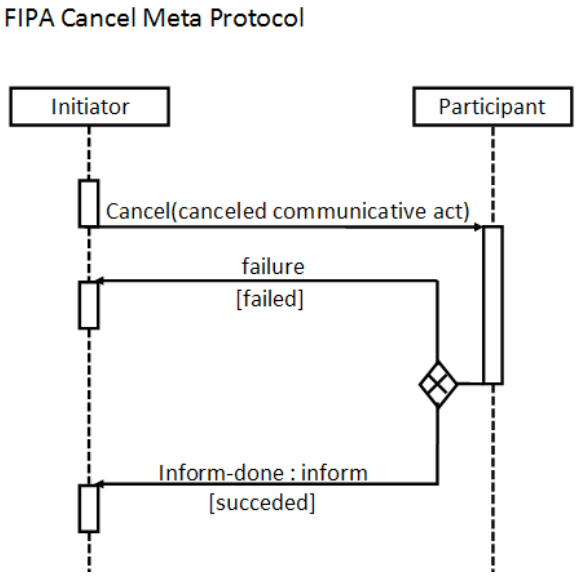

Figura 8 – Meta Protocolo FIPA-Cancel

Quando o participante recebe um *cancel* enviado pelo iniciador, só tem duas alternativas (para além de não perceber): informa o iniciador que a conversação especificada na mensagem *cancel* foi cancelada, usando a mensagem *inform* com conteúdo *done*...; ou informa o iniciador que o cancelamento falhou usando a mensagem *failure* com a razão da falha.

3.7 Protocolo FIPA-Contract-Net

O protocolo FIPA-Contract-Net é usado em negociações simples entre agentes, com dois papéis: iniciador e participante. Ao contrário do que se tem passado com os protocolos descritos até agora, o papel de participante pode ser desempenhado simultaneamente por vários agentes.

O protocolo FIPA-Contract-Net é inciciado pelo envio de uma mensagem *cfp* (*Call for Proposals*) solicitando a apresentação de propostas respeitando a condição especificada para a execução do serviço especificado (Figura 9). Os agentes a quem a mensagem *cfp* é enviada irão desempenhar o papel de participantes.

A mensagem iniciadora *cfp* deve especificar o limite temporal para a recepção de propostas através do parâmetro *:reply-by*.

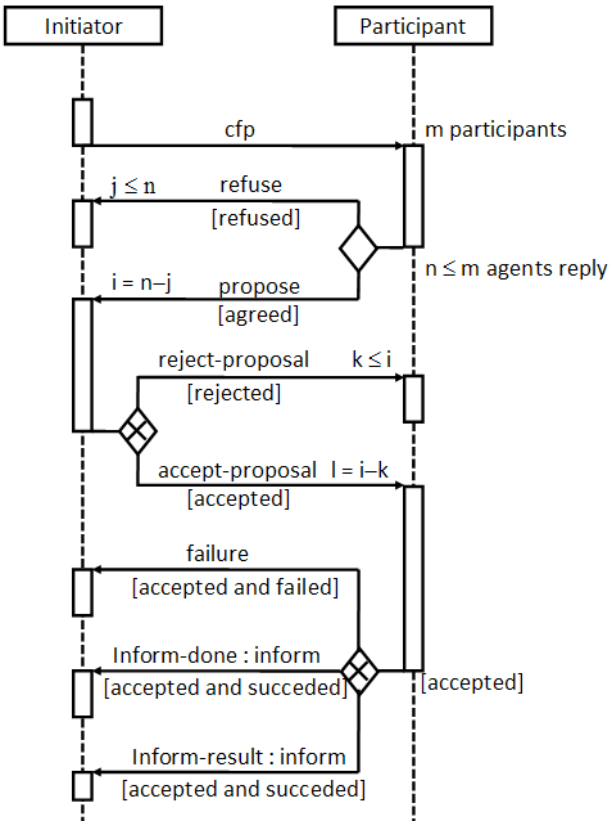

Figura 9 – Protocolo FIPA-Contract-Net

Quando recebe a mensagem *cfp*, o participante pode recusar participar na interacção, usando a mensagem *refuse* com a razão da recusa, terminando a conversação; ou apresentar uma proposta compatível com as condições especificadas, recorrendo à mensagem *propose*. Se o participante apresenta uma proposta, o seu papel no protocolo transita para o estado *proposed*.

Quando se atinge o limite temporal para a recepção de respostas dos participantes, o iniciador analisa e avalia as respostas obtidas. Se o iniciador recebe uma proposta de um dos participantes pode aceitar a proposta, usando a mensagem *accept-proposal* mediante uma condição de aceitação especificada na mensagem; ou rejeitar a proposta, usando a mensagem *reject-proposal* contendo a razão da rejeição, terminando assim esta instância do papel de participante no protocolo.

Todas as propostas recebidas pelo iniciador após o limite temporal para a sua recepção serão recusadas por essa razão.

Se a proposta do participante é aceite mediante uma condição de aceitação, a sua instância do papel de participante no protocolo transita para o estado *conditionally-accepted*. Neste estado, assim que a condição de aceitação for verdadeira, o participante fica obrigado a cumprir a proposta oferecida, transitando a sua instância do papel de participante no protocolo para o estado *accepted*.

Quando uma instância do papel de participante está no estado *accepted*, o agente associado a essa instância fica obrigado a executar o serviço proposto nas condições da propostas. Nessas circunstâncias, três coisas podem acontecer: a execução do serviço falha, caso em que o participante envia a mensagem *failure* coma a razão da falha; a execução tem sucesso resultando em informação para ser enviada, caso em que o participante envia uma mensagem *inform* com a informação resultante; ou a execução tem sucesso não resultando em informação a enviar, caso em que o participante envia uma mensagem *inform* dizendo que o serviço contratado foi executado com sucesso. Em qualquer destes casos, a intervenção deste agente participante no protocolo termina.

A intervenção do iniciador no protocolo termina apenas quando terminam as intervenções de todos os agentes participantes no protocolo.

Como em todas as conversações, todas as mensagens trocadas entre o iniciador e cada um dos participantes são identificadas através do parâmetro *:conversation-id* cujo valor é criado pelo iniciador

do protocolo quando envia a mensagem *cfp*. O iniciador é livre de criar um identificador de conversa diferente para cada um dos participantes ou usar sempre o mesmo.

Em qualquer momento, qualquer interveniente no protocolo pode não perceber uma mensagem, caso em que envia a mensagem *not-understood* com a razão de não ter percebido a mesnagem. Se isto acontecer a intervenção desse participante termina.

3.8 Protocolo FIPA-Iterated-Contract-Net

O protocolo FIPA-Iterated-Contract-Net (Figura 10) é uma versão iterada do protocolo FIPA-Contract-Net (secção 3.7).

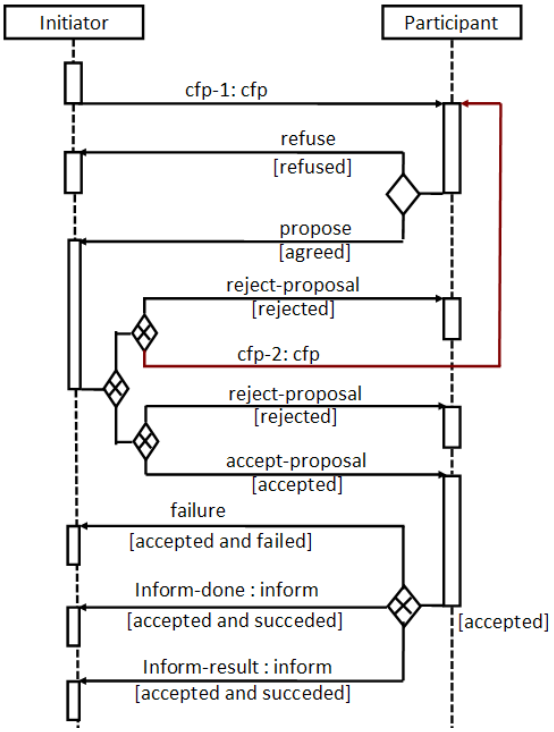

Figura 10 – Protocolo FIPA-Iterated-Contract-Net

Quando se atinge o tempo limite para apresentação de propostas, o iniciador avalia as propostas recebidas. Dessas, pode rejeitar algumas; aceitar algumas, caso em que o protocolo termina; ou voltar a enviar um *cfp* reformulado a alguns dos participantes que enviaram propostas. Neste caso, o protocolo repete-se. O FIPA-Iterated-Contract-Net termina quando todas as propostas são rejeitadas e o iniciador não envia um novo *cfp*, quando nenhuma proposta é apresentada, ou quando pelo menos uma proposta é aceite e o serviço contratado é prestado.

4 Referências Bibliográficas

[Bauer et al. 2001] Bauer, B.; Müller, J.P.; e Odell, J. 2001. "Agent UML: A Formalism for Specifying Multiagent Interaction". In Agent-Oriented Software Engineering, Paolo Ciancarini e Michael Wooldridge eds., Springer, Berlin

[Booch et al. 1999] Booch, G.; Rumbaugh, J.; e Jacobson, I. 1999. "The Unified Language User Guide". Addison-Wesley, Reading, MA

[FIPA 2001-31] Foundation for Intelligent Physical Agents. 2001. "FIPA English Auction Interaction Protocol Specification". Report 00031. http://www.fipa.org/specs/fipa00031/

[FIPA 2001-32] Foundation for Intelligent Physical Agents. 2001. "FIPA Dutch Auction Interaction Protocol Specification". Report 00032. http://www.fipa.org/specs/fipa00032/

[FIPA 2002-26] Foundation for Intelligent Physical Agents. 2002. "FIPA Request Interaction Protocol Specification". Report 00026. http://www.fipa.org/specs/fipa00026/

[FIPA 2002-27] Foundation for Intelligent Physical Agents. 2002. "FIPA Query Interaction Protocol Specification ".Report 00027. http://www.fipa.org/specs/fipa00027/

[FIPA 2002-28] Foundation for Intelligent Physical Agents. 2002. "FIPA Request When Interaction Protocol Specification". Report 00028. http://www.fipa.org/specs/fipa00028/

[FIPA 2002-29] Foundation for Intelligent Physical Agents. 2002. "FIPA Contract Net Interaction Protocol Specification". Report 00029. http://www.fipa.org/specs/fipa00029/

[FIPA 2002-30] Foundation for Intelligent Physical Agents. 2002. "FIPA Iterated Contract Net Interaction Protocol Specification". Report 00030. http://www.fipa.org/specs/fipa00030/

[FIPA 2002-33] Foundation for Intelligent Physical Agents. 2002. "FIPA Brokering Interaction Protocol Specification". Report 00033. http://www.fipa.org/specs/fipa00033/

[FIPA 2002-34] Foundation for Intelligent Physical Agents. 2002. "FIPA Recruiting Interaction Protocol Specification". Report 00034. http://www.fipa.org/specs/fipa00034/

[FIPA 2002-35] Foundation for Intelligent Physical Agents. 2002. "FIPA Subscribe Interaction Protocol Specification". Report 00035. http://www.fipa.org/specs/fipa00035/

[FIPA 2002-36] Foundation for Intelligent Physical Agents. 2002. "FIPA Propose Interaction Protocol Specification". Report 00036. http://www.fipa.org/specs/fipa00036/

[FIPA 2002-37] Foundation for Intelligent Physical Agents. 2002. "FIPA Communicative Act Library Specification". Report 00037. http://www.fipa.org/specs/fipa00037/

[Nowostawski et al. 2001] Nowostawski, M.; Purvis, M.; e Cranefield, S. 2001. "A Layered Approach for Modelling Agent Conversations". Proceedings of the Agents2001 2nd International Workshop on Infrastructure for Agents, MAS, and Scalable MAS. pp163-170

[Odell et al. 2000] Odell, J.; Parunak, H.V.D.; e Bauer, B. 2000. "Extending UML for agents", Proceedings of the AAAI2000 workshop on Agent-Oriented Information Systems. pp.3-17