

Tecnologia para Sistemas Inteligentes
Apontamentos para as aulas sobre
Interacção com o Directory Facilitator

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

1	DESCRIÇÃO DE SERVIÇOS	2
2	REGISTO DE UM SERVIÇO	4
3	CONSULTAR O DF	7
	3.1 EMPARELHAMENTO COM O PADRÃO DE PROCURA	7
	3.2 RESTRIÇÕES À PROCURA	8
	3.3 EXEMPLO DE UMA CONSULTA	8
4	REFERÊNCIAS BIBLIOGRÁFICAS	12

Interacção com o Directory Facilitator

O DF (*Directory Facilitator*) é um agente da plataforma FIPA que presta serviços de páginas amarelas. Qualquer agente que deseje anunciar serviços, pode registar os serviços no DF. Qualquer agente que deseje encontrar um agente que preste um dado serviço, pode usar o DF para identificar os prestadores do serviço desejado.

Tanto o registo de serviços como a procura de prestadores de serviços no DF recorrem à mensagem *request* enquadrada pelo protocolo FIPA-Request. O que varia é a acção que se pede ao DF para executar. No caso do registo de serviços, usa-se a acção *register* (registar). No caso da procura de prestadores de serviços, usa-se a acção *search* (procurar). O DF aceita outras acções (*modify* e *deregister*) cujos detalhes podem ser consultados em [FIPA 2002-23].

Cada plataforma de agentes tem que ter pelo menos um DF cuja identificação é

```
(agent-identifier
  :name df@hap_name
  :addresses (sequence hap_transport_address))
```

Nesta secção, ilustra-se a descrição de serviços, o registo de serviços no DF, e a procura de serviços no DF. Para isso, suporemos que um agente presta um serviço de informação sobre restaurantes, o qual irá ser registado no DF; e que outro agente pretende descobrir quem presta serviços de informação sobre restaurantes, consultando o DF. Primeiro que tudo, apresentaremos a forma como os serviços são descritos.

1 Descrição de Serviços

De acordo com as especificações da FIPA [FIPA 2002-23], para um agente registar os seus serviços no DF, é necessário usar um objecto da classe *df-agent-description*, a qual contém diversas informações genéricas sobre o agente que presta os serviços (e.g., o nome do agente), e um conjunto de descrições de serviços. Cada uma das descrições de serviço recorre a um objecto da classe *service-description*.

O registo de serviços no DF tem um prazo de validade para evitar que o DF tenha que manter indefinidamente descrições de serviços que já nem sequer existem. Se o agente que suporta um serviço pretender, poderá voltar a registar o serviço no DF quando tiver terminado o prazo de validade.

A classe *df-agent-description* está descrita na Tabela 1. A classe *service-description* está descrita na Tabela 2.

Parâmetro	Descrição	Tipo
name	Identificador do agente	agent-identifier
services	Conjunto dos services suportados pelo agente	Set of service-description
protocols	Conjunto de protocolos suportados pelo agente	Set of String
ontologies	Conjunto de ontologias conhecidas pelo agents	Set of String
languages	Conjunto de linguagens de conteúdo conhecidas pelo agente.	Set of String
lease-Time	Duração ou instante de tempo em que este registo expira.	datetime

Tabela 1 – Classe df-agent-description

Todos os atributos da classe *df-agent-description* são opcionais. O atributo *lease-Time* especifica o prazo de validade do registo ou o instante de tempo absoluto em que termina a sua validade.

Parâmetro	Descrição	Tipo
name	Nome do serviço	String
type	Tipo do serviço	String
protocols	Conjunto de protocolos de interacção associados ao serviço	Set of String
ontologies	Conjunto de ontologias associadas ao serviço	Set of String
languages	Conjunto de linguagens de conteúdo permitidas na interacção com o serviço	Set of String
ownership	Proprietário do serviço	String
properties	Conjunto de propriedades que descrevem o serviço	Set of property

Tabela 2 – Classe service-description

A classe *service-description*, usada para descrever um dado serviço tem um conjunto de atributos genéricos que se aplicam a qualquer tipo de serviço (e.g., name, type, and languages), e dispõe do atributo *properties* cujo valor é um conjunto de propriedades específicas de cada tipo de serviço. Cada propriedade é descrita através de um objecto da classe *property*, o qual se descreve na Tabela 3.

Atributo	Descrição	Tipo
Name	Nome da propriedade	String
value	Valor da propriedade, o qual pode ser qualquer	term

Tabela 3 – Classe property

A classe *property* representa um par propriedade-valor. Ambos os atributos da classe *property* são obrigatórios.

Um serviço de informação de restaurantes, usando a ontologia *Agentcities-Restaurant-Ontology* seria descrito pelo objecto representado na Figura 1.

```
(service-description
  :name restaurantes-de-lisboa
  :type information-service
  :protocols (set FIPA-query FIPA-subscribe)
  :ontologies (set Agentcities-Restaurant-Ontology)
  :languages (set OO-SL PCL)
  :ownership "we-be-mind::adetti"
  :properties (set
    (property :name location :value lisbon)
    (property :name update-frequency :value +00000000T24000000))
)
```

Figura 1 – Descrição do serviço de informação “restaurants-de-lisboa”

O serviço descrito chama-se “restaurantes-de-lisboa”, é da classe *information-service* e usa a ontologia de restaurantes definida no projecto *Agentcities*. Pode-se interactuar com este serviço (através do agente que o implementa), usando as linguagens de conteúdo OO-SL e PCL (Prolog Content language) e os protocolos de interacção FIPA-query e FIPA-subscribe.

O serviço é caracterizado por duas propriedades adicionais: a localização e a frequência de actualização.

2 Registo de um Serviço

Para que um agente se registre no DF, recorre à mensagem *request* em que o conteúdo é a acção *register*. O parâmetro passado à acção *register* é um objecto da classe *df-agent-description*, o qual contém um conjunto de objectos da classe *service-description* cada um dos quais descreve um serviço. A interacção de registo no DF é moldada pelo protocolo FIPA-request.

A mensagem mostrada na Figura 2 é enviada pelo agente *AgenteRestaurantesLisboa* ao DF da plataforma *PlataformaLisboa* para registar o seu serviço de informação sobre restaurantes.

```
(request
  :sender (agent-identifier :name AgenteRestaurantesLisboa)
  :receiver (set (agent-identifier
    :name df@PlataformaLisboa.we-b-mind.org
    :addresses (sequence http://PlataformaLisboa.we-b-mind.org/acc))
  )
  :content "(action
    (agent-identifier
      :name df@PlataformaLisboa.we-b-mind.org
      :addresses (sequence http://PlataformaLisboa.we-b-mind.org/acc))
    (register
      (df-agent-description
        :name (agent-identifier :name AgenteRestaurantesLisboa)
        :lease-Time +000000000T600000000T
        :services (set (service-description
          :name restaurantes-de-lisboa
          :type information-service
          :protocols (set FIPA-query FIPA-subscribe)
          :ontologies (set Agentcities-Restaurant-Ontology)
          :languages (set OO-SL PCL)
          :ownership "we-be-mind::adetti"
          :properties (set
            (property :name location :value lisbon)
            (property :name update-frequency :value +00000000T24000000))
          )
        )
      )
    ))"
  :reply-with register001
  :protocol fipa-request
  :ontologies (set fipa-agent-management)
  :language fipa-sl0
)
```

Figura 2 – Registo do agente Restaurantes no DF

O agente *AgenteRestaurantesLisboa* pede ao DF para responder usando a referência *register001*. A ontologia usada na mensagem é *fipa-agent-management* a qual é usada em todas as interacções com os agentes DF e AMS. A linguagem de conteúdo é SLO, uma versão mais simples da linguagem SL.

A descrição do agente *AgenteRestaurantesLisboa* não tem a especificação de todos os atributos porque estes são facultativos. A descrição do serviço é a mesma apresentada e explicada na secção 1.

De acordo com o protocolo FIPA-request, se o DF aceita efectuar o registo, enviará uma mensagem *agree* como a que se apresenta na Figura 3.

```
(agree
  :sender (agent-identifier
    :name df@PlataformaLisboa.we-b-mind.org
    :addresses (sequence http://PlataformaLisboa.we-b-mind.org/acc))
  :receiver (set
    (agent-identifier :name AgenteRestaurantesLisboa))
  :content "(action
    (agent-identifier
      :name df@PlataformaLisboa.we-b-mind.org
      :addresses (sequence http://PlataformaLisboa.we-b-mind.org/acc))
    (register
      (df-agent-description
        :name (agent-identifier :name AgenteRestaurantesLisboa)
        :lease-Time +000000000T600000000T
        :services (set (service-description
          :name restaurantes-de-lisboa
          :type information-service
          :protocols (set FIPA-query FIPA-subscribe)
          :ontologies (set Agenticities-Restaurant-Ontology)
          :languages (set OO-SL PCL)
          :ownership "we-be-mind::adetti"
          :properties (set
            (property :name location :value lisbon)
            (property :name update-frequency :value +00000000T240000000))
          )
        )
      )
    )
  )
  :in-reply-to register001
  :language fipa-sl0
  :protocol fipa-request
  :ontology fipa-agent-management
)
```

Figura 3 – O DF aceita registar o serviço de informação sobre restaurantes

O segundo elemento do conteúdo da mensagem *agree* é uma proposição que especifica as condições de aceitação do pedido de registo. Neste caso, é o símbolo proposicional *True* o qual, por ser sempre verdade por definição, é usado para indicar que a aceitação do pedido não depende de nenhuma condição.

Em vez da mensagem *agree* em que o DF aceita efectuar o registo, poderia ter enviado a mensagem *not-understood* se não tivesse compreendido a mensagem recebida, ou a mensagem *refuse* se não tivesse aceitado o registo. Em qualquer destes dois casos, o protocolo de interacção terminaria.

Após ter aceite o registo do serviço de informação sobre restaurantes do agente AgenteRestaurantesLisboa, o DF efectua o registo na sua base de dados. Se esta operação for executada com sucesso, o DF informa o AgenteRestaurantesLisboa que foi efectuado o registo (Figura 4).

```
(inform
  :sender (agent-identifier
    :name df@PlataformaLisboa.we-b-mind.org
    :addresses (sequence http://PlataformaLisboa.we-b-mind.org/acc))
  :receiver (set
    (agent-identifier :name AgenteRestaurantesLisboa))
  :content "(done (action
    (agent-identifier
      :name df@PlataformaLisboa.we-b-mind.org
      :addresses (sequence http://PlataformaLisboa.we-b-mind.org/acc))
    (register
      (df-agent-description
        :name (agent-identifier :name AgenteRestaurantesLisboa)
        :lease-Time +00000000T600000000T
        :services (set (service-description
          :name restaurantes-de-lisboa
          :type information-service
          :protocols (set FIPA-query FIPA-subscribe)
          :ontologies (set Agentcities-Restaurant-Ontology)
          :languages (set OO-SL PCL)
          :ownership "we-be-mind::adetti"
          :properties (set
            (property :name location :value lisbon)
            (property :name update-frequency :value +00000000T24000000))
          )
        )
      )
    )))"
  :in-reply-to register001
  :language fipa-sl0
  :protocol fipa-request
  :ontology fipa-agent-management
)
```

Figura 4 – O DF informa o agente que o seu serviço foi registado

Em vez de enviar a mensagem *inform-done*, o DF poderia ter enviado a mensagem *failure* se tivesse ocorrido uma falha durante a execução do registo do serviço na base de dados.

Agora o registo do serviço de informação de restaurantes está efectuado e será válido por um período de 60 horas. Se o agente de informação de restaurantes pretender actualizar o prazo de validade do registo do seu serviço, poderá enviar ao DF uma mensagem *request* com a acção *modify* para se alterar o valor do atributo *lease-Time* da descrição do agente.

Os prazos de validade dos registos no DF, quer sejam especificados durante o registo do serviço quer sejam especificados em alterações do registo subsequentes, são sujeitas a apreciação do DF. Se este aceitar o valor especificado pelo agente as mensagens de resposta resultantes do protocolo *fipa-request* (*agree* e *inform-done*) conterão um valor de *lease-Time* igual ao valor especificado. Se o DF não aceitar prazo de validade do registo proposto pelo agente, as mensagens protocolares de resposta (*agree* e *inform-done*) conterão o valor de *lease-Time* aceite pelo DF.

3 Consultar o DF

A consulta do DF, bem como a consulta do AMS, não utiliza as mensagens de pergunta do ACL (*query-ref*, *query-if*, e *subscribe*). Em vez de isso, usa-se a mensagem *request* com a acção especial *search*. A acção *search* recebe, no seu primeiro argumento, um padrão de procura. O segundo argumento da acção *search* é um objecto que descreve restrições à procura de informação, por exemplo o número máximo de resultados devolvidos.

O resultado da acção *search* é o conjunto de registos que emparelham com o padrão de procura. A resposta é uma mensagem *inform* em que o conteúdo é formado à custa do predicado *result*.

Infelizmente, este mecanismo de consulta é relativamente inflexível. Por exemplo, obriga o agente que consulta a receber toda a informação constante de cada registo que emparelha com o padrão de consulta, em vez de receber apenas os atributos necessários para a aplicação. Este mecanismo também não permite que se restrinja o número de registos recebidos através da especificação de condições relacionais com a forma de inequações, entre outras.

A próxima secção descreve a forma como a acção *search* usa o padrão de procura para encontrar registos que satisfazem esse padrão. Na secção seguinte é explicada a classe usada para especificar restrições de procura. Finalmente, é apresentado um exemplo de procura no DF.

3.1 Emparelhamento com o Padrão de Procura

O padrão de procura recebido na acção *search* é um objecto da mesma classe dos objectos procurados. Se a procura é feita no DF, em que os objectos procurados são descrições de agentes (*df-agent-description*), o padrão de procura é um objecto da classe *df-agent-description*.

De um modo geral, mas com excepções, um objecto emparelha com um padrão de procura se os seus atributos tiverem os valores especificados no padrão de procura. As excepções a esta regra são os atributos não especificados no padrão de procura e os atributos cujo valor é um conjunto ou uma sequência.

Antes de verificar o emparelhamento entre um objecto registado e um padrão de procura, a acção *search* avalia todas as expressões funcionais contidas no padrão de procura. Por exemplo, o valor 5 emparelha com o padrão (plus 2 3).

Um objecto registado emparelha com um padrão de procura se, após avaliação das expressões funcionais possivelmente contidas no padrão:

1. A classe do objecto for a mesma que a classe do padrão de procura;
2. Cada atributo do objecto registado satisfaz o padrão de procura no que diz respeito a esse atributo;
3. Um atributo do objecto registado satisfaz um padrão de procura se o padrão de procura não contiver a especificação desse atributo, ou se o padrão de procura especificar esse atributo e o valor do atributo do objecto registado emparelhar com o valor do atributo do padrão de procura;
4. Uma constante emparelha apenas com uma constante igual;
5. Uma sequência padrão vazia emparelha com uma sequência vazia; a sequência (cons x sequência1) emparelha com a sequência padrão (cons y sequência2) se x emparelhar com y e sequência1 emparelhar com a sequência padrão sequência2, ou se sequência1 emparelhar com a sequência padrão (cons y sequência2); e
6. Um conjunto padrão emparelha com um conjunto se cada elemento do conjunto padrão for emparelhado por pelo menos um elemento do conjunto.

Por exemplo, a sequência padrão (sequence a d) emparelha com a sequência (sequence a b c d e) mas não com a sequência (sequence b c d e a), e o conjunto padrão (set a c b) emparelha com o conjunto

(set a b c d). E, num exemplo mais complexo, o padrão (O :a va :b (set v3 v2) :c (C :ca vca)) emparelha com (O :a va :b (set v1 v2 v3) :c (C :ca vca :cb vcb) :d vd).

3.2 Restrições à Procura

Os resultados e o tipo de procura efectuada pela acção *search* podem ser condicionados através do seu segundo argumento, o qual especifica um conjunto de restrições à procura. A classe *search-constraints* é usada para especificar restrições à procura.

Parâmetro	Descrição	Tipo
max-depth	Profundidade máxima da procura em serviços de directoria federados. Valores negativos indicam que a procura federada não tem limites	integer
max-results	Número máximo de resultados a obter como resultado da procura. Um número negativo indica que o número de resultados não tem limites.	integer
search-id	Identificador único da operação de procura	String

Tabela 4 – Classe search-constraints

Todos os atributos da classe *search-costraints* são opcionais.

3.3 Exemplo de uma consulta

Nesta secção apresenta-se uma interacção em que o agente dunny consulta o DF para encontrar um agente que suporte o serviço de informação de restaurantes em localizado onde quer que seja e com uma frequência de actualização de 24 horas (+00000000T24000000).

```

(request
  :sender (agent-identifier
    :name dummy@foo.com
    :addresses (sequence iiop://foo.com/acc))
  :receiver (set (agent-identifier
    :name df@PlataformaLisboa.we-b-mind.org
    :addresses (sequence http://PlataformaLisboa.we-b-mind.org/acc)))
  :content
    "(action
      (agent-identifier
        :name df@PlataformaLisboa.we-b-mind.org
        :addresses (sequence http://PlataformaLisboa.we-b-mind.org/acc))
      (search
        (df-agent-description
          :services (set
            (service-description
              :type information-service
              :ontologies (set Agentcities-Restaurant-Ontology)
              :properties (set
                (property :name update-frequency :value +00000000T24000000))))
          (search-constraints
            :max-results 1))))"
  :reply-with query003
  :language fipa-sl0
  :protocol fipa-request
  :ontology fipa-agent-management
)

```

Figura 5 – Procura no DF

A mensagem da Figura 5 ilustra a falta de flexibilidade do mecanismo de procura definido para o DF e para o AMS. Por exemplo, não seria possível restringir os resultados aos identificadores de agentes que prestassem um serviço de informação de restaurantes em que a periodicidade de actualização fosse igual ou inferior a 24 horas.

Se o DF aceitar este pedido, enviará a mensagem *agree* e efectuará a procura, após o que em caso de sucesso, enviará uma mensagem *inform* como a que se apresenta na Figura 6.

```

(inform
  :sender
    (agent-identifier
      :name df@PlataformaLisboa.we-b-mind.org
      :addresses (sequence http://PlataformaLisboa.we-b-mind.org/acc))
  :receiver (set (agent-identifier
    :name dummy@foo.com
    :addresses (sequence iiop://foo.com/acc)))
  :content
    "(result
      (action
        (agent-identifier
          :name df@PlataformaLisboa.we-b-mind.org
          :addresses (sequence http://PlataformaLisboa.we-b-mind.org/acc))
        (search
          (df-agent-description
            :services (set
              (service-description
                :type information-service
                :ontologies (set Agentcities-Restaurant-Ontology)
                :properties (set (property
                  :name update-frequency :value +00000000T24000000))))
            (search-constraints :max-results 1)))
        (set (df-agent-description
          :name (agent-identifier :name AgenteRestaurantesLisboa)
          :lease-Time +00000000T600000000T
          :services (set (service-description
            :name restaurantes-de-lisboa
            :type information-service
            :protocols (set FIPA-query FIPA-subscribe)
            :ontologies (set Agentcities-Restaurant-Ontology)
            :languages (set OO-SL PCL)
            :ownership "we-be-mind::adetti"
            :properties (set
              (property :name location :value lisbon)
              (property :name update-frequency :value +00000000T24000000)) )))
          )))
    )"
  :in-reply-to query003
  :language fipa-s10
  :protocol fipa-request
  :ontology fipa-agent-management
)

```

Figura 6 – Resposta a uma procura no DF

A informação encontrada pelo DF como resultado da consulta efectuada é um conjunto de objectos da classe *df-agent-description*. Estes resultados são enviados numa mensagem *inform* em que o conteúdo

é uma proposição criada com o predicado *result*, o qual é usado para representar o resultado da execução de acções informativas. *Result* tem dois argumentos: o primeiro é um termo de acção, e o segundo é o resultado da sua execução.

4 Referências Bibliográficas

[FIPA 2002-23] Foundation for Intelligent Physical Agents. 2002. “FIPA Agent Management Specification”. Report 00023. <http://www.fipa.org/specs/fipa00023/>