

## 4 Framework DRP-MAS

Nesta seção, descrevemos o framework DRP-MAS (Diagnosing and Recommending Plans on Multi-Agent Systems) responsável por realizar diagnósticos e prover recomendações para agentes que não atingiram algum objetivo. Inicialmente, a idéia geral do framework é apresentada, seguida pela explicação da arquitetura, e seus respectivos módulos, além da maneira com que o framework oferece suporte para aplicações baseadas em computação ubíqua. Logo após são apresentados os diagramas de caso de uso com suas respectivas descrições e os diagramas de seqüência. Para finalizar é apresentada uma abordagem alternativa de como aplicar os conceitos propostos pelo DRP-MAS a partir de diferentes tecnologias.

### 4.1. Visão Geral

O framework DRP-MAS visa ajudar um agente que não alcançou um dos seus objetivos após realizar a execução de um plano. Este agente, denominado agente Solicitante, requisita a um agente Mediador um agente Diagnóstico. Quando o Mediador recebe a mensagem de solicitação, ele cria um agente Diagnóstico (responsável por realizar diagnósticos), e um agente de Recomendação (responsável por prover recomendações). Em seguida envia uma mensagem ao Solicitante informando qual agente Diagnóstico irá trabalhar de forma exclusiva para o agente que o solicitou (**Figura 9**).

A partir do momento que o Solicitante conhece o seu agente Diagnóstico, ele pode requisitar recomendações que o permitam alcançar o objetivo desejado (**Figura 10**). Com essa finalidade, o Solicitante envia uma mensagem para o agente Diagnóstico, composta por um conjunto de informações sobre a execução realizada que serão úteis tanto para gerar o diagnóstico como também prover as recomendações. Algumas dessas informações enviadas pelo Solicitante podem ser as seguintes: plano executado, objetivo desejado, perfil do agente Solicitante e um número que represente a qualidade da execução realizada pelo plano (maiores detalhes na subseção 4.2.2). Esse conceito de

qualidade é baseada nos trabalhos [Horling et al., 2000], [Wagner et al., 2003] e [Horling et al., 2007].

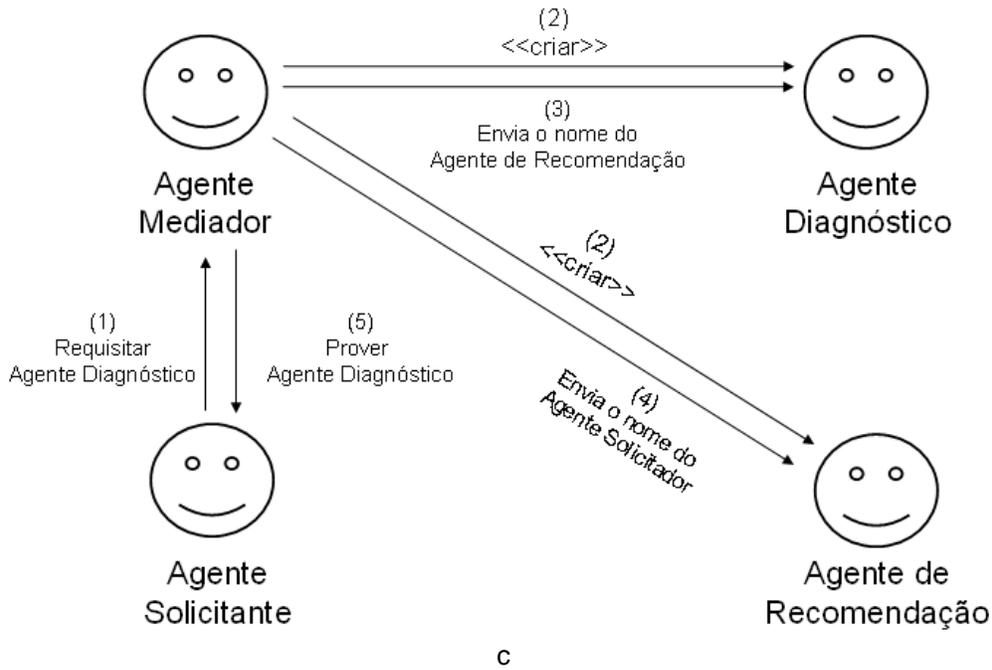


Figura 9. Modelo Conceitual para requisitar agente Diagnóstico.

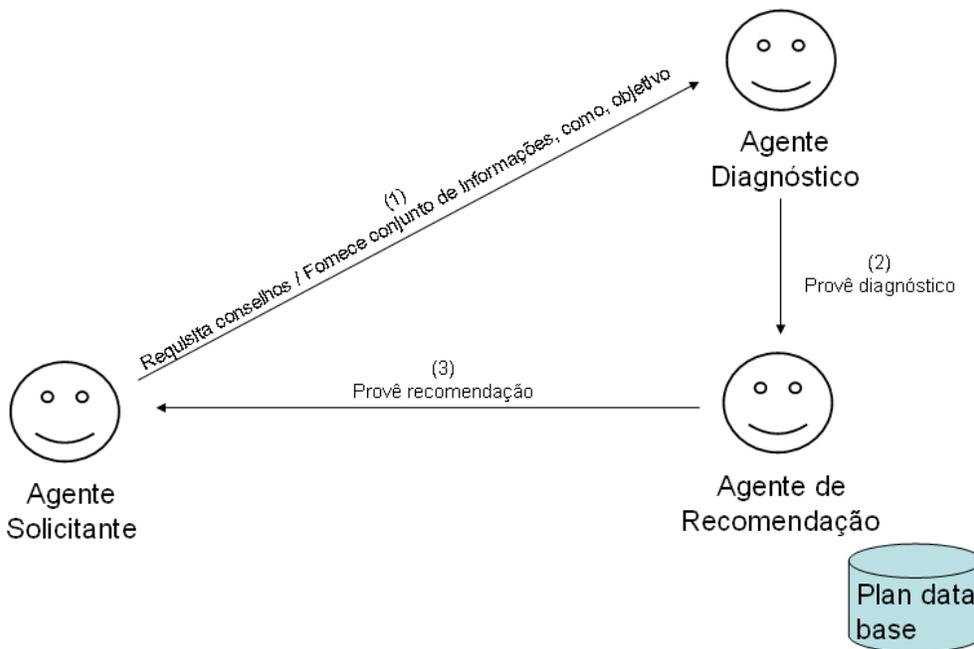


Figura 10. Modelo conceitual para solicitar recomendações

Quando o agente Diagnóstico recebe a mensagem, ele procura encontrar a razão que impediu o Solicitante de não alcançar algum objetivo. Quando a análise é finalizada, ele fornece o diagnóstico para o agente de Recomendação. Mesmo que não seja encontrado algum diagnóstico, o agente Diagnóstico envia uma mensagem para o agente de Recomendação informando que não foi possível encontrar o motivo que impediu o Solicitante de alcançar o objetivo desejado. Neste caso, o agente de Recomendação simplesmente seleciona outros planos que permitam alcançar o mesmo objetivo para que em seguida eles sejam fornecidos ao Solicitante.

Na situação que algum diagnóstico seja encontrado, o agente de Recomendação procura planos alternativos de execução a partir do diagnóstico encontrado e do objetivo que o agente Solicitante deseja alcançar. Quando o diagnóstico indica que ocorreu um problema em alguma negociação com outro agente, uma análise é feita para concluir quais agentes podem ser aconselhados em futuras interações. Tanto a estratégia de recomendação, como a de diagnóstico são pontos flexíveis no framework.

A partir de um conjunto de agentes que possam participar nas negociações com o Solicitante, o agente de Recomendação usa o valor de reputação destes agentes para selecionar os “melhores”, i.e., aqueles que possuem as melhores reputações. O perfil do agente Solicitante pode ser uma importante informação para definir quais agentes devem ser selecionados. Ao terminar a execução do agente Recomendação, uma mensagem é fornecida ao agente Solicitante com as recomendações selecionadas, i.e. com os planos de recomendados.

Para representar a idéia do uso de reputação com os processos de diagnóstico e recomendação, foram definidos dois modelos de reputação: centralizado e descentralizado. Detalhes são apresentados na subseção 4.2.6.

Antes que um Solicitante requisiite alguma recomendação a partir dos agentes do DRP-MAS o agente de Recomendação tem que conhecer quais são os possíveis planos que poderá recomendar para cada agente da aplicação. Não somente os planos, mas também informações que estejam relacionados com eles, como, por exemplo, possíveis diagnósticos relacionados e objetivos que o plano pode alcançar são informações que o agente de Recomendação deve receber (seção 4.2.5). Esses planos são informados por agentes do sistema e são usados pelos agentes de Recomendação para fornecer conselhos aos agentes Solicitantes.

## 4.2. Arquitetura

Nesta seção, descrevemos a arquitetura da nossa abordagem. Como ilustrado na **Figura 11**, o framework DRP-MAS é composto por cinco módulos: Mediação, Diagnóstico, Recomendação, Artificial Intelligence Toolkit e Reputação.

O módulo de Mediação define o agente Mediador, responsável por criar um agente Diagnóstico e um agente de Recomendação para um Solicitante definido na Aplicação. O módulo de Diagnóstico realiza o processo de diagnóstico, enquanto que o módulo de Recomendação provê recomendações para que algum objetivo possa ser alcançado. O módulo Artificial Intelligence Toolkit usa uma API (Application Public Interface) chamada Bigus [Bigus, 2001], que permite o uso de diferentes algoritmos de raciocínio (reasoning) nos processos de diagnóstico e de recomendação. Os algoritmos providos são os seguintes: forward chaining, backward chaining e lógica fuzzy. E finalmente o módulo de Reputação, responsável por representar o conceito reputação.

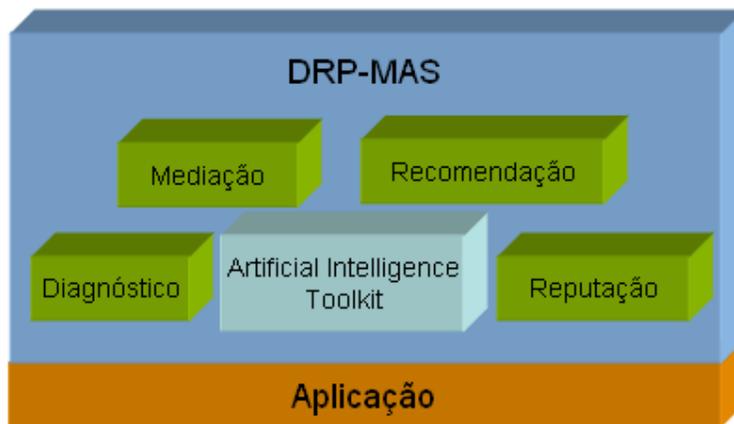


Figura 11. Arquitetura DRP-MAS.

O framework foi implementado utilizando-se JADEX (ver subseção 2.1), considerado uma das mais úteis plataformas para criar agentes de software. Essa plataforma foi escolhida por prover uma implementação para uma arquitetura abstrata de Belief-Desire-Intention (BDI) [Braubach et al., 2004], que é a base para nossa abordagem de diagnósticos e recomendações. Essa abordagem é baseada em falhas que acontecem enquanto agentes tentam alcançar seus objetivos (desejos) a partir da execução de planos (intenções).

Como o JADEX foi o framework escolhido, cada agente do DRP-MAS possui um arquivo XML, responsável por descrever uma série de informações,

como, por exemplo, quais planos serão executados por cada um, qual sua base de conhecimento (belief base), quais mensagens o agente poderá receber ou enviar, além de quais objetivos pretende alcançar. Assim, agentes como, Mediador, Diagnóstico, e de Recomendação possuem um XML que descreve o template (esqueleto) de cada agente oferecido pelo DRP-MAS. A instância do framework deve somente fazer uma cópia do template correspondente, e em seguida preencher os pontos flexíveis.

Para facilitar o entendimento dos módulos definidos no DRP-MAS, estes são explicados em mais detalhes nas seções seguintes junto com a descrição de quais informações podem compor o conjunto de dados enviados por um agente Solicitante para auxiliar a encontrar diagnósticos e selecionar recomendações.

#### **4.2.1. Módulo Mediação**

Este módulo é composto por um agente Mediador, responsável por criar um agente Diagnóstico e um agente de Recomendação a partir de alguma requisição realizada por um agente Solicitante. Após a criação dos agentes, o Mediador informa o nome do agente de Recomendação para o agente Diagnóstico, e em seguida o nome do agente Solicitante para o de Recomendação.

A abordagem proposta permite que diversos agentes Mediadores sejam definidos, e que cada um forneça um agente Diagnóstico e um agente de Recomendação exclusivo para o Solicitante. Adotando esta abordagem evitamos que os agentes Solicitantes fiquem esperando em longas filas de resposta. Esta solução é válida sempre e quando não viole a carga suportada pelo sistema.

Outro ponto relevante sobre nossa abordagem é a razão que nos motivou a definir dois agentes de software no lugar de somente um para realizar tanto o processo de diagnóstico como o de recomendação. Com esta abordagem é possível combinar diferentes estratégias de diagnóstico com diferentes estratégias de recomendação.

Na **Figura 12** ilustramos a idéia em que mediadores criam combinações de agentes Diagnóstico e de Recomendação com diferentes estratégias a partir de solicitações. Percebe-se que o Mediador A fornece ao Solicitante A um agente Diagnóstico com uma estratégia diferente do fornecido pelo Mediador B. No entanto, ambos criam agentes de Recomendação com a mesma estratégia.

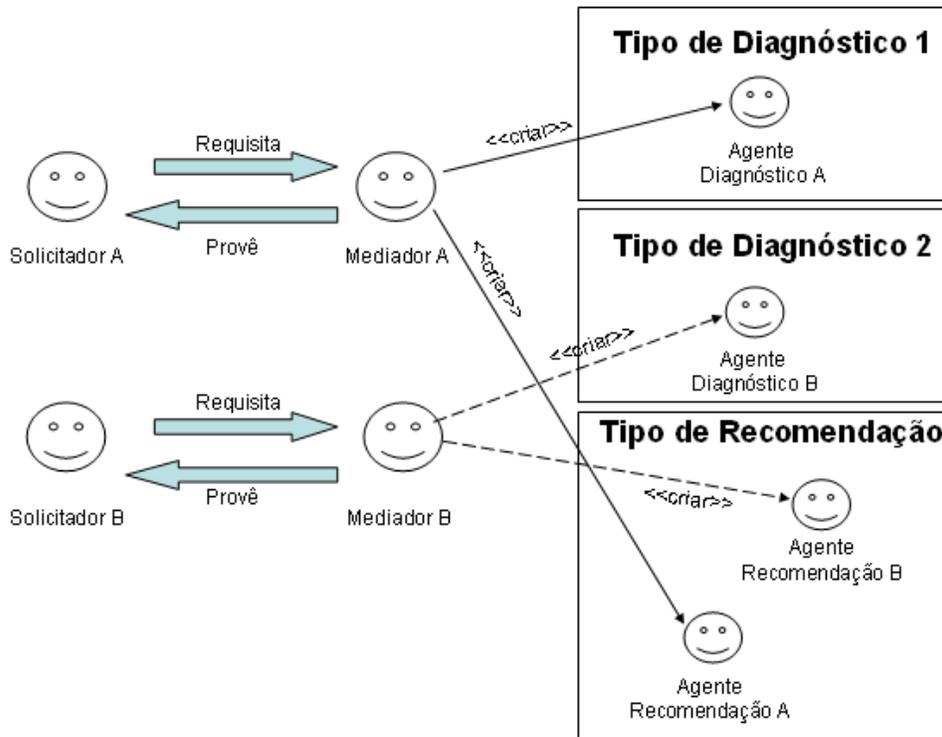


Figura 12. Mediadores criando diferentes tipos de agentes.

Para definir quais agentes serão criados por um agente Mediador, duas classes (ver **Figura 13**) devem ser estendidas: *PlanRegisterService* e *PlanCreateAgents*. Na primeira classe, o método abstrato *getServiceDescription* deve ser implementado para indicar qual serviço será oferecido pelo agente Mediador. O registro desse serviço é realizado no Directory Facilitator (serviços de páginas amarelas) do JADEX a partir do método *body* (já implementado pelo framework). Dessa forma, os agentes Solicitantes conseguem descobrir quais agentes são criados por cada agente Mediador do sistema.

Na classe *PlanCreateAgents* o método abstrato *createAgents* deve ser implementado. Nele é realizada a criação de um agente Diagnóstico e de um agente de Recomendação.

Após a criação dos planos, devem ser indicadas as respectivas classes no arquivo XML do agente Mediador correspondente. Na **Tabela 1** é apresentado um exemplo contendo as classes: *PlanInstanceRegisterService* e *PlanInstanceCreateAgents*. A primeira classe, responsável por realizar o registro do serviço provido pelo agente, é executada automaticamente logo após a execução do Mediador. Já a segunda classe é executada somente quando uma requisição é feita por um agente Solicitante.

Os planos *PlanRegisterService* e *PlanCreateAgents* aplicam o padrão Template Method, já que o método `body`, utiliza os métodos abstratos implementados pelas instâncias do framework.

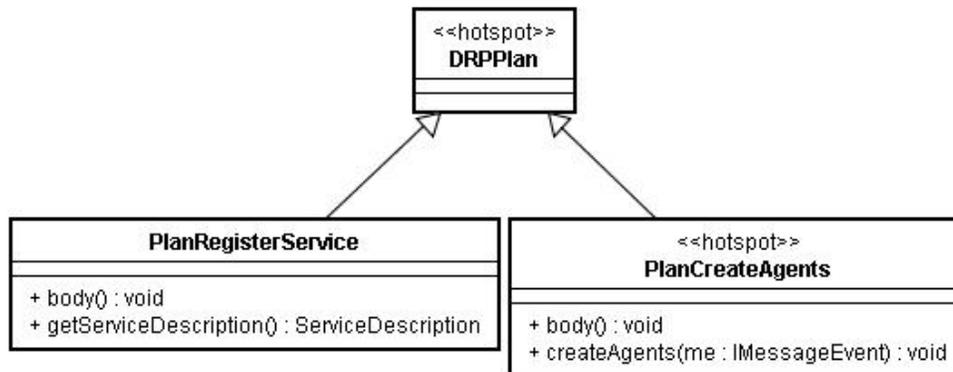


Figura 13. Diagrama de classe do módulo mediação.

```

<plan name="registerService">
  <body>new br.pucrio.example.planMediator.PlanInstanceRegisterService() </body>
</plan>
<plan name="createAgents">
  <body>new br.pucrio.example.planMediator.PlanInstanceCreateAgents() </body>
  <trigger>
    <messageevent ref="request_nameDiagnosis"/>
  </trigger>
</plan>
  
```

Tabela 1. Definir plano de criação de agente no XML do agente Mediador.

#### 4.2.2.

### Conjunto de Dados para Diagnósticos e Recomendações

Quando o agente Solicitante realiza a execução de um plano, mas seu objetivo não é alcançado, este poderá pedir o diagnóstico e a recomendação de novos planos. Para ajudar no processo de diagnóstico e recomendação, o agente poderá fornecer um conjunto de informações úteis sobre sua execução.

O DRP-MAS define um conjunto de informações que podem ser fornecidas pelos Solicitantes. A seguir, cada um desses dados é explicado, e na **Tabela 2** eles são listados acompanhados por uma breve descrição.

1. Recursos e problemas associados – Em [Horling et al., 2000] recursos são considerados dados importantes para realizar diagnósticos. Uma possível razão para impedir o sucesso na execução de um plano pode ser a ausência ou uma quantidade insuficiente de um recurso.

2. Qualidade do serviço – Definir um grau de qualidade para um plano executado por um agente, assim como definido no modelo TAEMS [Wagner et al., 2003] [Lesser et al., 2007], pode ser útil em diagnósticos. Esse grau pode representar diferentes informações, como, por exemplo, indicar se todos os passos executados foram bem sucedidos.
3. Objetivo – A execução de um plano sempre está associada a um objetivo, que algum agente deseja alcançar [Silva et al., 1999]. Conhecê-lo é fundamental para fornecer um diagnóstico e seus correspondentes conselhos. Tais conselhos, responsáveis por indicar quais planos alternativos executar, são a base para alcançar o objetivo referente.
4. Plano executado – Torna-se necessário conhecer o plano executado pelo agente Solicitante para que seja compreendida a razão responsável pela sua falha, assim como prover caminhos alternativos que permitam o alcance do objetivo desejado.
5. Agentes com quem o Solicitante interagiu – O diagnóstico pode indicar que um agente não forneceu de forma correta um serviço solicitado. Logo, torna-se importante conhecer quais agentes estavam envolvidos durante a execução o plano do agente Solicitante.
6. Serviços usados – Os serviços utilizados durante a execução do agente podem ser úteis para ajudar quais agentes recomendar em futuras interações. Conhecer os serviços e as características dos serviços utilizados pelo agente é fundamental não só para fornecer o diagnóstico como também para ajudar na recomendação de futuros parceiros que possam fornecer de maneira mais apropriada os serviços causadores da falha.
7. Perfil – Cada agente tem um perfil, que pode ser usado para representar algumas de suas características. O perfil do agente Solicitante pode estipular, por exemplo, o grau mínimo de reputação dos parceiros com os quais deseja interagir. Essa informação pode ser útil no processo de prover recomendações realizadas pelo agente de Recomendação, onde a escolha segue o perfil do agente Solicitante.
8. Base de Belief – A base de conhecimento usada pelo agente Solicitante mais a informação da sua última atualização podem ser úteis tanto para o processo de diagnóstico como o de recomendação.

Dados	Breve Descrição
Recursos	Recursos usados durante alguma execução.
Qualidade do serviço	Número que define a qualidade da execução de um plano. Idéia baseada no modelo TAEMS.
Objetivo	Objetivo não alcançado.
Plano executado	Plano executado pelo agente Solicitante e que não alcançou o objetivo desejado.
Agentes negociados	Nome dos agentes requisitados durante a execução do plano.
Serviços usados	Serviços usados pelo plano.
Perfil	Perfil do agente Solicitante.
Base de Belief	Base de conhecimento do agente.

Tabela 2. Conjunto de dados que define a qualidade de serviço

#### 4.2.3. Módulo Diagnóstico

O processo de diagnóstico mencionado na seção 4.1 é realizado pelo agente Diagnóstico provido pelo framework DRP-MAS. Tais diagnósticos são baseados nos dados providos pelo agente Solicitante (ver seção 4.2.2). Devido à possibilidade de representar diferentes estratégias de diagnóstico, o DRP-MAS definiu esse ponto como um hot spot [Fayad et al., 1999], isto é, um ponto flexível do framework.

Visando ilustrar uma possível estratégia, focamos em um simples domínio de fazer café. Consideremos um agente que tem como objetivo fazer café para alguns amigos, e que para alcançá-lo ele executa um plano qualquer. Suponhamos que o agente notificou que o café não está bom, mas não entende o por quê. Existem diversas razões que podem a uma má produção, como, por exemplo, a qualidade do pó de café, a temperatura da água ou a inadequada quantidade de pó ou de água usada. Para determinar o ocorrido, o agente Solicitante deve enviar um conjunto de dados para o agente Diagnóstico, i.e., (i) seu objetivo (fazer café para três pessoas), (ii) o plano executado para fazer o café, (iii) o café resultante, (iii) a quantidade de água, (iv) a temperatura da água e a (v) qualidade e a (vi) quantidade do pó de café usado. O agente diagnóstico deverá conhecer as seguintes informações: (i) o quanto de água e pó de café é necessário para fazer uma xícara de café, (ii) a temperatura ideal para a água, (ii) além da qualidade de várias pós de café. Para ajudar com a implementação

das estratégias, três algoritmos (backward chaining, forward chaining e lógica fuzzy) estão disponíveis no módulo de Inteligência Artificial (IA).

Um diagnóstico encontrado pode ser classificado como Main Diagnosis ou Inferred Diagnosis (ver **Figura 14**). O Main Diagnosis indica quais diagnósticos foram encontrados a partir das informações providas pelo agente Solicitante, e o Inferred Diagnosis representa aqueles diagnósticos encontrados a partir de alguma inferência, dada a escassa informação fornecida pelo Solicitante (subseção 4.2.4.). Assim, quando um processo de diagnóstico é finalizado, vários diagnósticos podem ser detectados.

Para definir uma estratégia de diagnóstico, basta estender a classe *PlanDiagnosing* (aplica o padrão Template Method), implementar o método *performingDiagnosis*, e em seguida definir o plano no arquivo XML do agente Diagnóstico. O exemplo apresentado na **Tabela 3** indica que o plano criado foi *PlanInstanceDiagnosis*.

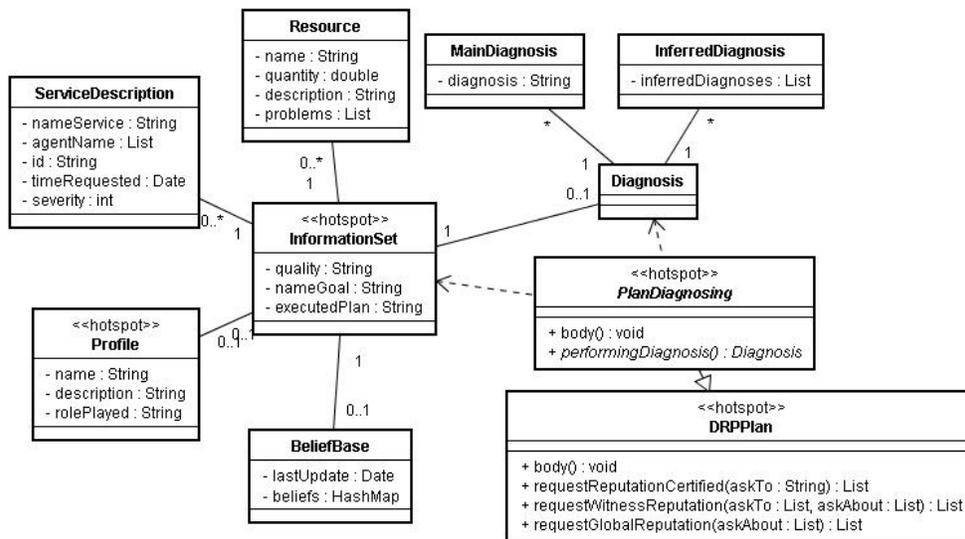


Figura 14. Diagrama de classes do processo de diagnóstico.

```

<plan name="diagnostic">
  <body>new br.pucrio.example.PlanInstanceDiagnosis() </body>
  <trigger>
    <messageevent ref="request_diagnosis"/>
  </trigger>
</plan>
    
```

Tabela 3. Definir plano no XML do agente *Diagnóstico*.

#### 4.2.4. Módulo de Inteligência Artificial

Como mencionado em seções anteriores e ilustrado na **Figura 15**, o módulo de Inteligência Artificial oferece três algoritmos de IA que podem ser usados por estratégias definidas pelas instâncias do framework. Os algoritmos são os seguintes: backward chaining, forward chaining e lógica fuzzy.

O framework estende uma API oferecida por [Bigus and Bigus, 2001] para que as instâncias tivessem maior facilidade em reusá-la, além de oferecer o serviço de inferência em diagnósticos (melhor explicado a frente). Na **Figura 16** são apresentadas as principais classes que compõe o módulo proposto. Perceba que para cada algoritmo de IA há uma classe que estende a classe abstrata *IA*, responsável por definir os métodos *init* e *importVariables* e que devem ser implementados pelas suas classes concretas. O primeiro método deve conter a base de regras do algoritmo selecionado. Nesse caso, classes oferecidas pela Bigus são usadas, como as classes *BooleanRuleBase* e *FuzzyRuleBase*. Já o segundo método, importa os valores que são analisados na base de regras. A partir dos valores especificados, em particular no algoritmo *Forward Chaining*, verifica-se quais variáveis na base de regras foram efetivamente usadas em alguma execução. Essa verificação é realizada para que o serviço de inferência de diagnósticos possa ser usado. As classes *UsedVariable* e *NotUsedVariable* do módulo de IA foram definidas para representar os dados utilizados e não utilizados, respectivamente.

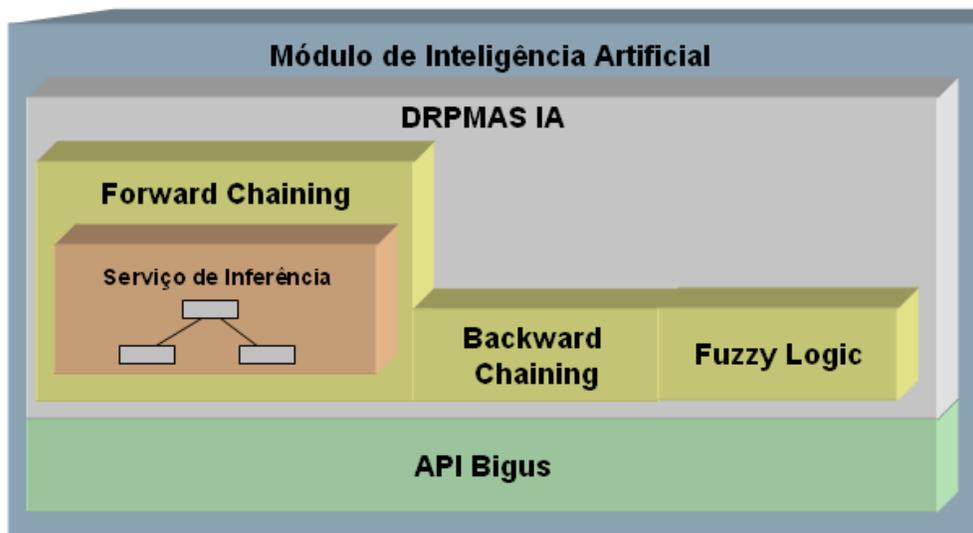


Figura 15. Arquitetura do módulo de Inteligência Artificial.

Embora o framework ofereça 3 algoritmos diferentes, nosso foco principal está no algoritmo forward chaining. Acreditamos que essa abordagem seja muito útil para realizar diagnósticos, já que visa usar inferência de regras para que, a partir de um conjunto de dados disponíveis, possa extrair mais dados enquanto procura atingir algum objetivo. Essa abordagem parece apropriada para o caso em que um agente Solicitante provê um conjunto de dados para um agente Diagnóstico que utilizará uma base de regras para concluir diagnósticos e fornecer para o agente de Recomendação. Uma base de regras define um conjunto de regras capaz de sugerir diferentes diagnósticos se baseando no conjunto de informação recebido. Quando mais informação for enviada pelo agente Solicitante, mais preciso será o diagnóstico.

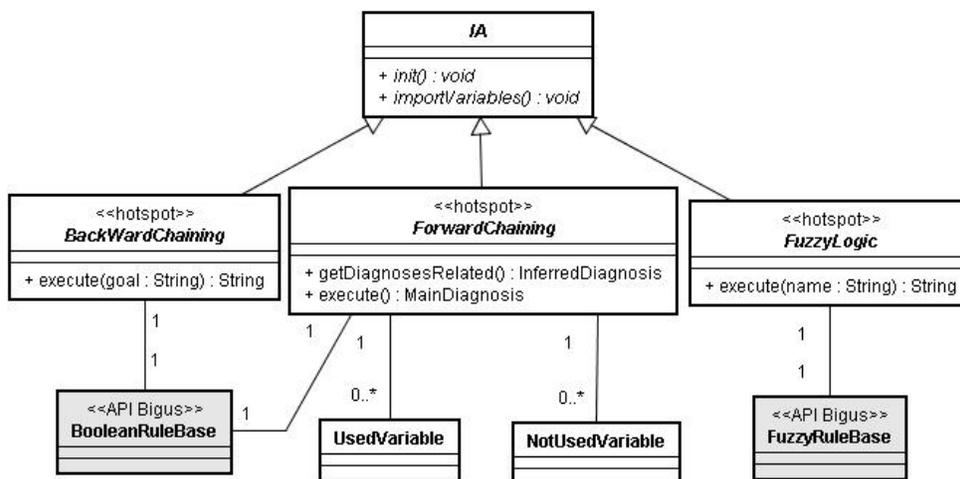


Figura 16. Diagrama de Classes do módulo de Inteligência Artificial.

Nos casos onde pouca informação é fornecida, diagnósticos pouco precisos serão encontrados. Porém, o framework é capaz de sugerir diagnósticos mais precisos a partir de suposições sobre as informações que não foram enviadas. Como descrito na seção anterior, o agente Diagnóstico é então capaz de fornecer dois tipos de diagnósticos: diagnósticos encontrados a partir das informações fornecidas e diagnósticos inferidos.

A árvore de diagnósticos ilustrada na **Figura 17** será utilizada como exemplo para explicar os dois tipos de diagnósticos. Imagine que o Diagnóstico A seja encontrado a partir de um conjunto de dados fornecidos pelo Solicitante. O framework verifica quais outros diagnósticos poderiam ser inferidos caso mais dados tivessem sido fornecidos pelo Solicitante, neste caso diagnósticos B, C, D e E. Primeiramente, o DRP-MAS analisa os dados usados para encontrar o Diagnóstico A, e usando o algoritmo forward chaining, ele verifica quais dados

não foram providos e quais poderiam ser usados para encontrar novos diagnósticos. Diagnósticos mais precisos são aqueles que mais se aproximam das folhas da raiz. Em nossa abordagem, consideramos diagnósticos irmãos aqueles que têm a mesma altura na árvore de diagnósticos.

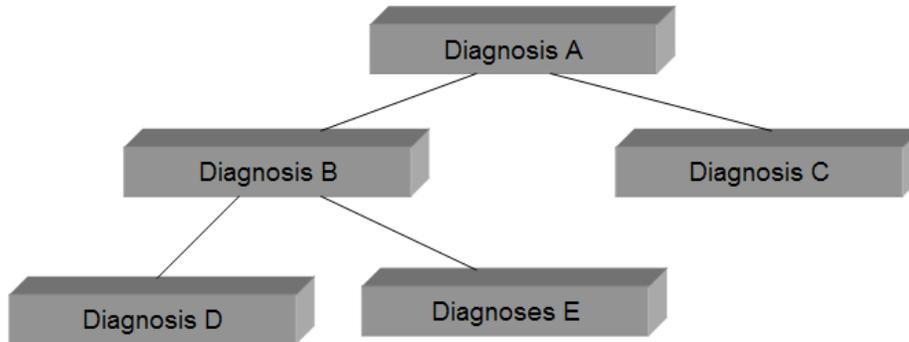


Figura 17. Relação de diagnósticos

Para exemplificar a idéia apresentada, imagine que um agente Solicitante deseje a tradução de uma palavra em português para inglês e peça este serviço a um agente tradutor. Quando o agente tradutor não consegue realizar a tradução com sucesso, o Solicitante solicita recomendações ao agente Diagnóstico. Quando o agente Diagnóstico recebe a requisição, ele usa o algoritmo forward chaining para inferir os diagnósticos a partir dos dados providos pelo Solicitante. Imagine que o Solicitante informou: (i) a qualidade da execução do serviço (igual a zero quando nenhuma tradução é realizada), (ii) a palavra a ser traduzida, (iii) e o dicionário usado para realizá-la. Após receber tais informações, o agente Diagnóstico checa se a palavra está no dicionário, e a seguir usa a base de regras, apresentada na **Tabela 4**, para inferir o diagnóstico. Caso a palavra não seja encontrada no dicionário do agente Tradutor, e o diagnóstico inferido seja *Error\_Translation*, isso significa que não foi identificado em detalhes qual erro aconteceu durante a tradução.

Se o agente Solicitante tivesse provido mais informações para o agente Diagnóstico, dois outros diagnósticos poderiam ser encontrados, assim como ilustrado na **Figura 18**. Um dos motivos para que a palavra solicitada não fosse traduzida, poderia ser: (i) a palavra ter sido escrita de forma incorreta ou (ii) ter sido utilizado um dicionário antigo para a tradução (não atualizado). Dado os dois possíveis diagnósticos mais detalhados, o agente de Recomendação recebe além do diagnóstico real, encontrado a partir dos dados fornecidos pelo Solicitante, os outros dois diagnósticos que poderiam ser encontrados caso mais informações tivessem sido fornecidas pelo agente Solicitante.

<pre> <b>IF</b> Quality_of_Service=0 <b>AND</b> Base_Belief=FALSE <b>THEN</b> DIAGNOSIS=ERROR_TRANSLATION  <b>IF</b> Quality_of_Service=0 <b>AND</b> Base_Belief =FALSE <b>AND</b> Base_Belief_Updated=FALSE <b>THEN</b> DIAGNOSIS=NECESSARY_REQUEST_WORD  <b>IF</b> Quality_of_Service=0 <b>AND</b> Base_Belief =FALSE <b>AND</b> Base_Belief_Updated=TRUE <b>THEN</b> DIAGNOSIS=NOT_POSSIBLE_TO_TRANSLATE </pre>
--

Tabela 4. Base de regras para tradução de inglês para português.

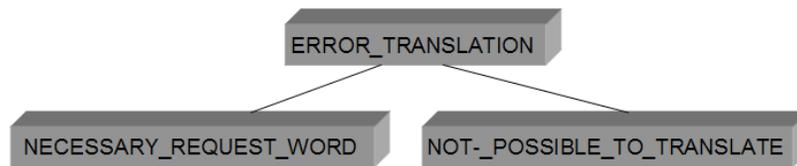


Figura 18. Relação entre diagnósticos definida na base de regras da tradução.

#### 4.2.5. Módulo de Recomendação

Este módulo possui um agente de Recomendação responsável por prover caminhos alternativos para que um agente Solicitante possa alcançar algum objetivo desejado. Este processo é composto por três passos (ver **Figura 19**):

1. Selecionando planos
2. Verificando planos selecionados
3. Selecionando agentes

O primeiro passo é executado quando o agente de Recomendação recebe o diagnóstico enviado pelo agente Diagnóstico e precisa verificar quais planos podem ser utilizados para alcançar um objetivo específico. Baseado nas informações providas pelo agente Solicitante, o agente de Recomendação analisa os planos disponíveis em uma base, para que assim possa escolher aqueles que serão recomendados.

O segundo passo verifica se os planos selecionados implicam em interações com outros agentes. Caso os planos não pressuponham interações, o

processo é concluído, caso contrário, é realizada uma solicitação de agentes para futuras colaborações ao agente de Reputação (detalhes na seção 4.2.6). Em seguida o terceiro passo é executado. No terceiro passo, o agente de Recomendação recebe os agentes candidatos escolhidos pelo agente de Reputação a partir de suas reputações e, em seguida, seleciona quais agentes recomenda para quais planos. Ao final, os planos e os agentes selecionados são informados para o agente Solicitante. As subseções seguintes realizam explicações mais detalhadas para cada passo mencionado, além de uma descrição de como criar uma instância desse processo.

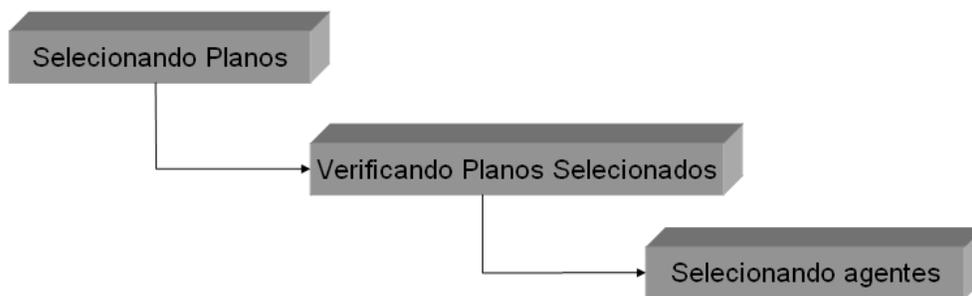


Figura 19. Processo de execução do agente Recomendação.

#### 4.2.5.1. Selecionando Planos

Esse passo é responsável por escolher planos alternativos para alcançar o objetivo desejado. Como cada aplicação pode desenvolver diferentes estratégias, o framework DRP-MAS define cada estratégia como um hot spot, que pode ser instanciado pela aplicação.

Para que seja possível selecionar planos baseados nas informações fornecidas a partir de um agente Solicitante, cada plano deve estar associado a um conjunto de informações que o descrevem e classificam. Torna-se importante definir os serviços e os recursos usados durante a execução, o objetivo que o plano tentará alcançar, diagnósticos relacionados, além de outras informações. Observe que as informações associadas com cada plano são fortemente relacionadas com as informações que podem ser providas pelos possíveis agentes Solicitantes da aplicação (ver subseção 4.2.2.).

O DRP-MAS oferece dois tipos de serviço que ajudam na definição da estratégia de recomendação. Um dos serviços permite selecionar planos que são relacionados com informações fornecidas e o outro serviço permite

selecionar planos que não estão relacionados com um determinado conjunto de dados. Após selecionar os planos para recomendação, o segundo passo do processo de recomendação deve ser executado. Na situação em que nenhum plano seja selecionado para recomendação, uma mensagem é enviada para o Solicitante e o processo é abortado.

#### **4.2.5.2. Verificando Planos**

Após a seleção dos planos alternativos, esse passo verifica se os planos irão precisar de outros agentes para fornecer certos serviços. Se nenhum plano precisa solicitar serviços, uma mensagem é enviada para o agente Solicitante com as recomendações obtidas. Caso contrário, o agente de Recomendação terá que encontrar agentes confiáveis que possam prover os serviços necessários. Para selecionar esses agentes confiáveis o agente de Recomendação solicita ao agente de Reputação (subseção 4.2.6) um conjunto de agentes candidatos a partir dos serviços necessários e do perfil do agente Solicitante.

#### **4.2.5.3. Selecionando Agentes**

Esse passo é responsável por selecionar os agentes que serão recomendados a partir de uma seleção provida pelo agente de Reputação. Caso algum plano não tenha agentes para recomendar, o plano é descartado, no entanto, se tiver agentes para aconselhar o plano é recomendado. A estratégia para definir quais agentes recomendar é outro hot-spot do framework, já que diferentes modelos de reputação podem ser usados, além de diferentes perfis. Ao final da execução, os planos são recomendados ao agente Solicitante e o agente de Recomendação aguarda outra solicitação.

#### **4.2.5.4. Instanciando Processo de Recomendação**

A **Figura 20** ilustra parte das classes que compõe o módulo de Recomendação. Nela são apresentados dois planos, *PlanSelectingPlans* e *PlanChoosingAgents* que aplicam o padrão Template Method. O primeiro plano representa os dois primeiros passos do processo de recomendação: selecionando planos e verificando planos. Para definir a estratégia do primeiro

passo, a instância do framework deve implementar o método abstrato *selectingPlans*. Esse método deve utilizar a base de planos (classe *PlanBase*) que possui os planos fornecidos por outros agentes do sistema. Nessa base são identificados os dados relacionados com cada plano: (i) recursos necessários para alguma execução, (ii) objetivo, (iii) diagnósticos que o plano pode solucionar, (iv) serviços oferecidos por outros agentes e necessários na execução, (v) perfis aceitos de agentes para executar o plano e (vi) uma base de conhecimento. Para representar a relação dessas informações com cada plano, foi definida a classe *TemplatePlan*, que é usada como base para gerar as recomendações finais, representada pela classe *RecommendedPlan*.

O plano *PlanChoosingAgents*, também apresentado na **Figura 20**, representa o terceiro passo do processo de recomendação, Selecionando Agentes. Para definir a estratégia de seleção, basta a instância do framework DRP-MAS implementar o método *verifyRecommendedAgents*. Ao final da execução do plano, uma mensagem contendo uma instância da classe *Recommendation* associada com instâncias da classe *RecommendedPlan* é enviada ao Solicitante correspondente. Instâncias da classe *RecommendedPlan* possuem informações que não estão presentes em instâncias do *TemplatePlan*, como, por exemplo, os agentes que serão usados para prover os serviços necessários e as informações detalhadas dos recursos necessários (ex: a quantidade do recurso).

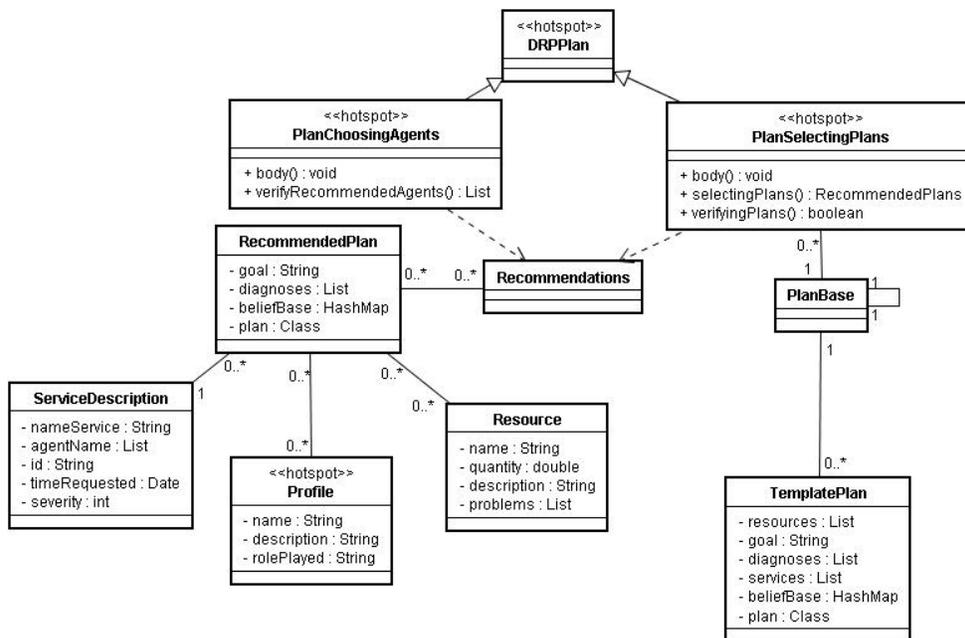


Figura 20. Diagrama de classes do processo de recomendação.

Na **Tabela 5** é ilustrada parte de um arquivo XML de um agente de Recomendação, responsável por informar quais planos da instância serão executados no processo de recomendação. No nosso exemplo as classes dos planos seriam: *PlanInstanceSelectingPlan* e *PlanInstanceChoosingAgents*.

```
<plan name="selectingPlan">
  <body>new br.pucrio.example.PlanInstanceSelectingPlan()</body>
  <trigger>
    <messageevent ref="inform_diagnosis"/>
  </trigger>
</plan>
<plan name="choosingAgents">
  <body>new br.pucrio.example.PlanInstanceChoosingAgents()</body>
  <trigger>
    <messageevent ref="inform_reputation"/>
  </trigger>
</plan>
```

Tabela 5. Definição de planos em um XML de um agente de Recomendação.

#### 4.2.6. Módulo de Reputação

O módulo de reputação é o responsável por representar o conceito reputação de agentes no DRP-MAS. O framework oferece dois tipos de modelos: centralizado e descentralizado. O primeiro modelo, baseado no framework REPORT (ver subseção 2.4.1), representa a situação em que diversos agentes compartilham suas opiniões sobre outros agentes em uma base de reputação que armazena a reputação global de cada agente. Já o segundo modelo baseado no modelo Fire (ver subseção 2.4.2), permite que agentes criem suas próprias reputações sem a necessidade de compartilhá-las com outros agentes, isto é, cada agente pode ter sua própria base de reputações. A **Figura 21** ilustra a idéia mencionada.

Os tipos de reputação oferecidos pelo framework são os seguintes: global, interaction trust, witness e certified. A primeira reputação faz parte do modelo centralizado, enquanto que as outras fazem parte do modelo descentralizado. A seguir, os modelos são explicados em detalhes.

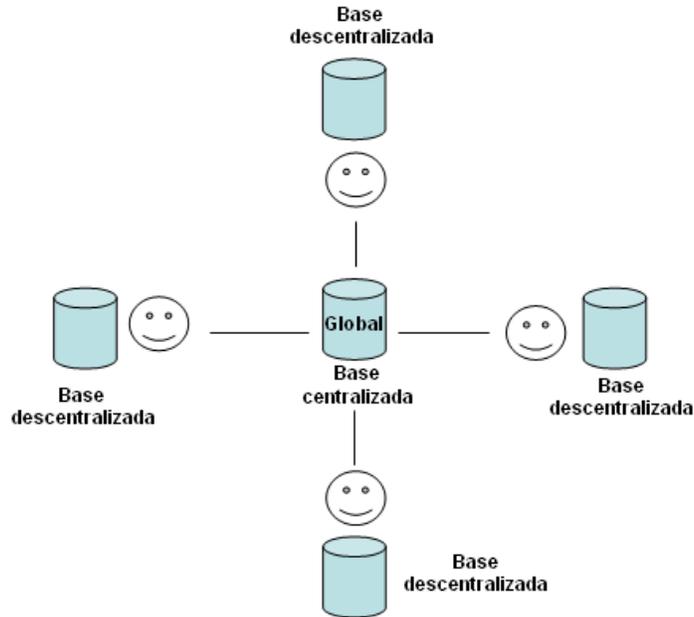


Figura 21. Modelos de reputação, centralizado e descentralizado.

#### 4.2.6.1. Modelo Centralizado

O modelo centralizado, assim como mencionado no início da subseção 4.2.6, utiliza o conceito de reputação global proposto pelo framework REPORT. Essa reputação é representada pela classe *GlobalReputation* apresentada na **Figura 22**. Considerando essa abordagem, cada agente poderá ter uma reputação única baseada nas informações providas por outros agentes do sistema. Tais reputações são armazenadas em uma base de reputação definida pela classe *CentralisedReputationBase*, que aplica o padrão Singleton [Gamma et al., 1994].

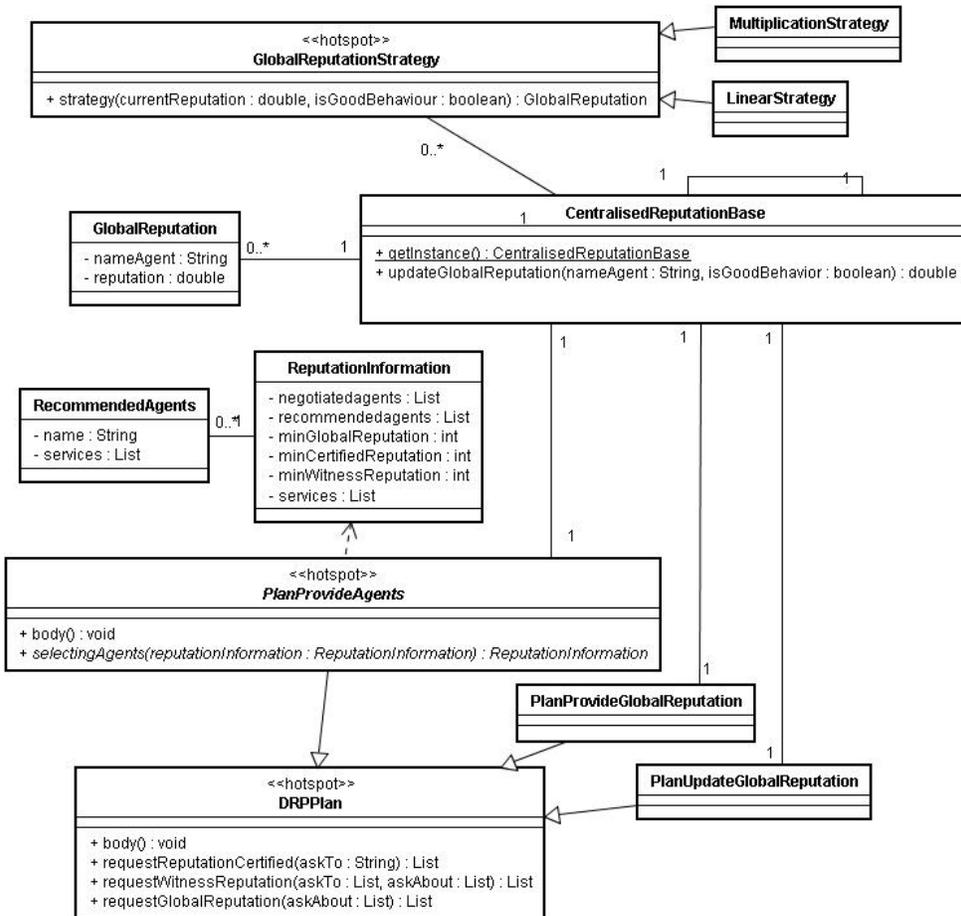


Figura 22. Diagrama de classe do modelo centralizado.

Visando auxiliar o acesso às reputações, um agente de Reputação foi definido no framework. Esse agente é o responsável por realizar todas as atualizações das reputações globais, além de fornecê-las aos agentes solicitantes.

Pode-se perceber na **Figura 22** que diferentes estratégias podem ser definidas para calcular as reputações globais a partir da classe *GlobalReputationStrategy*, que aplica o padrão Strategy [Gamma et al., 1994]. O framework oferece duas estratégias padrões que podem ser usadas pelas instâncias: soma (classe *LinearStrategy*) e multiplicação (classe *MultiplicationStrategy*). A primeira estratégia incrementa a reputação global de um agente baseada na informação fornecida por outro agente. Esse dado fornecido informa se o respectivo agente comportou-se bem ou não em interações passadas. Caso tenha se comportado bem, a reputação é incrementada em 0.1 (fórmula 1) até chegar ao valor um, caso contrário, a reputação é decrementada em 0.1 (fórmula 2).

$$\text{repGlobal} = \text{repGlobal} + 0.1 \quad (1)$$

$$\text{repGlobal} = \text{repGlobal} - 0.1 \quad (2)$$

A estratégia de multiplicação segue a mesma idéia da estratégia de soma, no entanto, no lugar de incrementar por 0.1, multiplica-se por 1.1 (fórmula 3). A partir do momento em que a reputação global supera o valor um o valor um é atribuído. No caso em que o agente tenha se comportado mal, a reputação global é dividida por 1.1 (fórmula 4).

$$\text{repGlobal} = \text{repGlobal} * 1.1 \quad (3)$$

$$\text{repGlobal} = \text{repGlobal} / 1.1 \quad (4)$$

Na **Figura 22** também são apresentados três planos que são executados pelo agente de Reputação. Tais planos são representados pelas classes: *PlanUpdateGlobalReputation*, *PlanUpdateGlobalReputation* e *PlanProvideGlobalReputation*. O primeiro plano é responsável por atualizar a reputação global de agentes do sistema. O segundo fornece reputações globais solicitadas por outros agentes, e já o terceiro plano recomenda um conjunto de agentes para um agente de Recomendação (ver subseção 4.2.5). Essa recomendação é baseada nas seguintes informações: (i) a reputação mínima aceitável pelo agente Solicitante para negociar com outros agentes, (ii) uma lista de serviços que os agentes devem fornecer, e (iii) a lista de agentes com os quais o agente Solicitante já interagiu. A partir dessas informações representadas pela classe *ReputationInformation*, o plano pode indicar quais agentes serão recomendados e quais serviços cada agente pode oferecer. Para aplicar a estratégia de seleção, basta implementar o método *selectingAgents* oferecido pela classe *PlanProvideAgents* (aplica o padrão Template Method). Como padrão, o DRP-MAS oferece uma seleção de agentes baseada nos serviços desejados e a reputação global mínima. Por isso, o template do arquivo XML do agente de Reputação, oferecido pelo DRP-MAS, inclui um plano chamado *PlanAgentReputation* (ver **Tabela 6**). Caso uma instância do framework deseje usar outro plano para a seleção de agentes candidatos, basta informá-lo no lugar do plano indicado.

Perceba que a classe *DRPPLAN*, responsável por definir o conceito plano no framework, oferece alguns métodos, dentre eles o *requestGlobalReputation*, que solicita reputações globais para o agente de Reputação. Dessa forma, qualquer agente do sistema que esteja executando algum plano poderá em qualquer instante solicitar tais reputações.

```
<plan name="provideReputation">
  <body>new DRPMAS.subsystem.reputation.plan.PlanAgentReputation()</body>
  <trigger>
    <messageevent ref="request_reputation_master"/>
  </trigger>
</plan>
```

Tabela 6. Parte do XML do agente de Reputação.

#### 4.2.6.2. Modelo Descentralizado

Este modelo é baseado no modelo Fire (ver subseção 2.4.2), responsável por definir um mecanismo de reputação descentralizado a qual os agentes avaliam o comportamento dos outros agentes, além de armazenarem as reputações definidas. Dentre as diferentes reputações propostas pelo modelo Fire, o DRP-MAS usa as seguintes:

- interaction trust (resultante de experiências passadas em interações realizadas);
- witness reputation (testemunhos sobre o comportamento dos agentes);
- certified reputation (cada agente armazena as referências providas por outros agentes sobre o seu comportamento).

O motivo principal que nos levou a escolher o modelo Fire, diferentemente de outros modelos de reputação, é o uso da reputação certificada. Tal reputação é fundamental quando um agente deseja conhecer a reputação de outro agente, com o qual ele nunca tenha interagido, e quando ele não conhece os agentes que já negociaram com o respectivo. Basta perguntar ao agente de interesse por suas reputações certificadas.

A **Figura 23** ilustra o modelo descentralizado utilizado pelo framework DRP-MAS. A partir da classe *DecentralisedReputation* outras reputações, além das três reputações mencionadas, podem ser definidas. E utilizando-se a classe *DecentralisedReputationStrategy* (que implementa o padrão Strategy) é possível especificar diferentes estratégias de cálculo para as reputações. Como padrão, o framework DRP-MAS oferece o cálculo proposto pelo modelo Fire (classe *StandardStrategy*).

Como cada agente pode calcular suas próprias reputações, considerando o modelo abordado, uma base descentralizada foi representada pela classe *DescentralisedReputationBase*. Assim, quando um agente deseja conhecer a reputação de outro agente calculada por um terceiro agente, basta solicitar ao agente que a possui. Para facilitar essas solicitações, a classe DRPPLAN fornece dois métodos apresentados na **Figura 22**: *requestReputationCertified* e *requestWitnessReputation*. O primeiro método solicita as reputações certificadas, enquanto o segundo método solicita as reputações de testemunho. Tanto a reputação certificada como a de testemunho são calculadas a partir do *interaction trust*, que é uma reputação visível somente ao agente que a calcula.

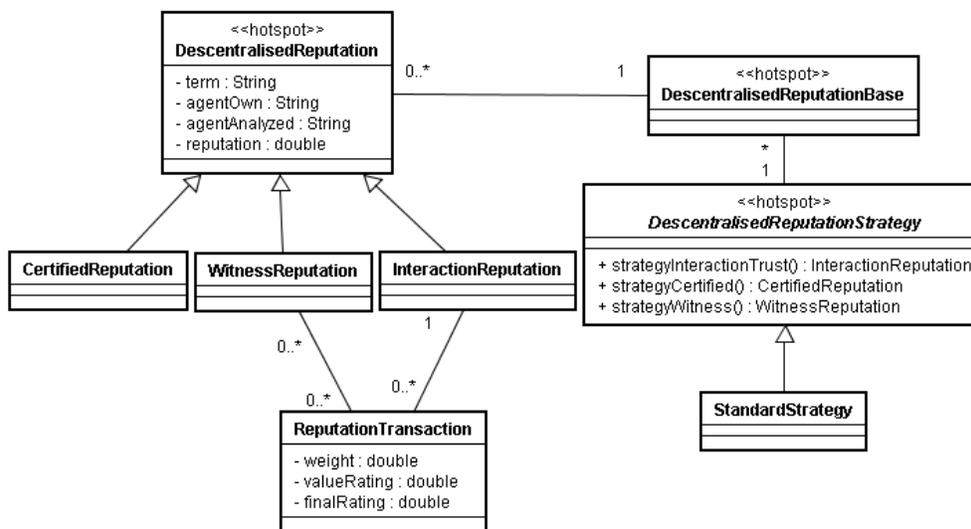


Figura 23. Diagrama de classe do modelo descentralizado.

Assim como no modelo centralizado, o agente de Reputação também pode utilizar as reputações definidas no modelo descentralizado para recomendar agentes ao agente de Recomendação (ver subseção 4.2.5). Além da reputação global mínima ilustrada no diagrama de classes da **Figura 22** e mencionada na subseção 4.2.6.1, as reputações certificada e de testemunho também podem ser usadas para selecionar agentes. A estratégia definida pelo agente de Reputação define quais informações serão usadas para selecionar os agentes.

### 4.3. Fornecendo Suporte para Computação Ubíqua

Weiser introduziu a área de computação ubíqua definindo um mundo onde computadores iriam prover informações e serviços quando e onde fosse

necessário. Essa proliferação realmente aconteceu a partir do avanço tecnológico de diferentes dispositivos, indo desde portáteis, como PDAs, celulares e laptops, até grandes dispositivos compartilhados.

O comportamento dos agentes de software que executam em aplicações baseadas em computação ubíqua freqüentemente se adapta ao tipo de dispositivo onde os agentes estão executando. A maneira como um determinado serviço deve ser provido tende a se adaptar ao dispositivo onde o agente está executando, já que dispositivos diferentes possuem características e limitações diferentes. Se um agente envia uma informação para um determinado dispositivo, talvez este dispositivo não seja capaz de interpretar o dado recebido como, por exemplo, alguns dispositivos possuem limitações no tipo de formato de arquivo que pode ler.

Preocupado em oferecer suporte para a realização de diagnósticos e recomendações para esses tipos de aplicações baseadas em computação ubíqua, o framework DRP-MAS define além do conjunto de informações mencionado na subseção 4.2.2 os seguintes dados:

1. Dispositivo usado – Devido às diferentes características dos dispositivos disponíveis, torna-se importante identificar: (i) o tipo de dispositivo usado (ex: celular, PDA, laptop, etc), (ii) o modelo do dispositivo (ex: LG MG296 GSM, Motorola Razr V3 Black GSM, etc.) e (iii) a linguagem em que os dados têm que ser providos pelo agente (ex: inglês, espanhol, português, etc).
2. Conexões – As características das conexões, i.e., (i) sua velocidade (ex: 56Kbps, 512Kbps, 3Mbps, etc), (ii) sua tecnologia (ex: wireless, LAN, WAN, etc.) e (iii) o endereço IP são importantes para construir um diagnóstico e prover recomendações.

Na Figura 24 podemos ver duas classes chamadas Device e Connection referentes às informações supracitadas. Perceba que elas podem ser usadas no processo de recomendação, assim como ilustrado na Figura 25.

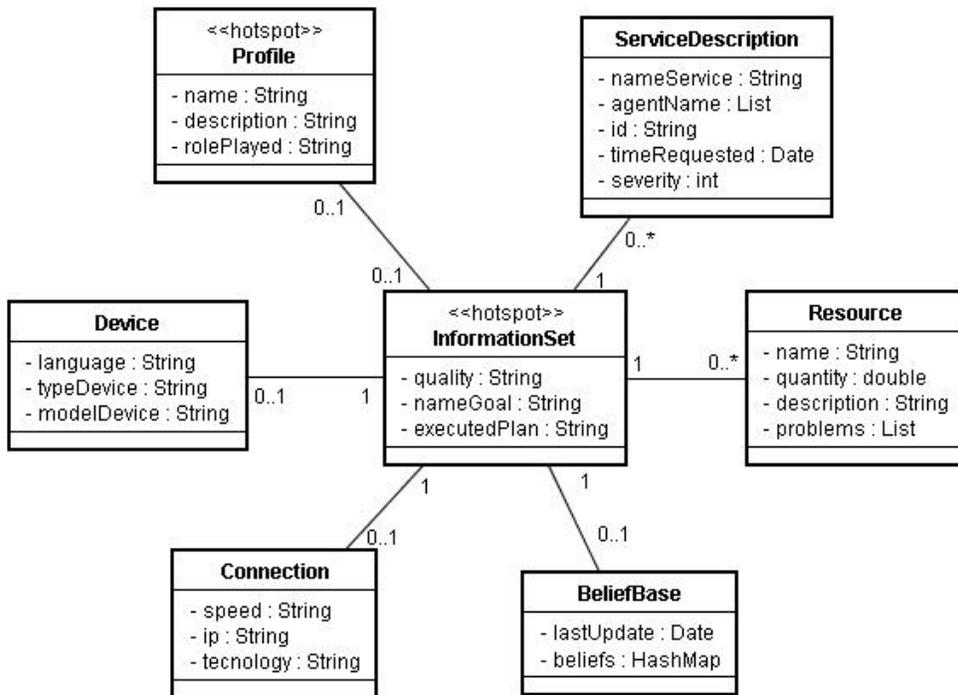


Figura 24. Conjunto de dados providos pelo agente Solicitante no domínio de computação ubíqua.

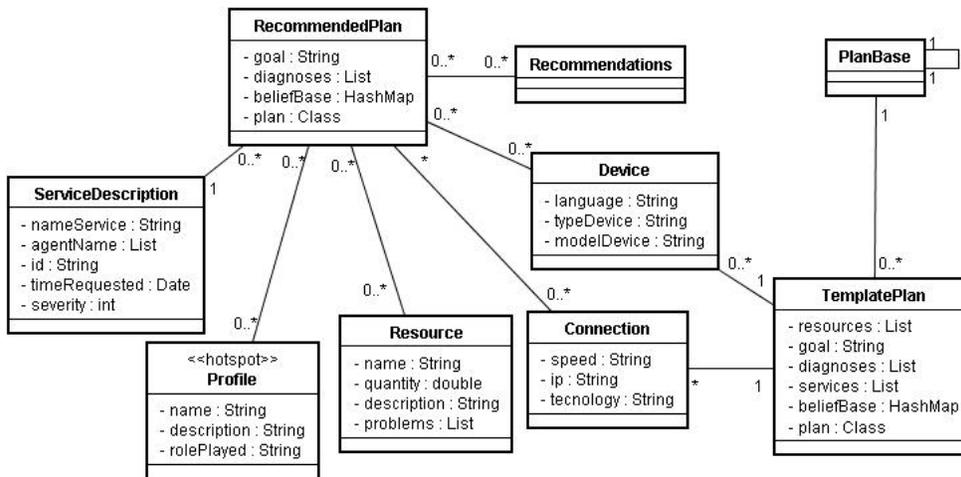


Figura 25. Dados de computação ubíqua usados no processo de recomendação.

#### 4.4. Pontos Flexíveis do Framework

Como já mencionado nas subseções anteriores, existem cinco pontos flexíveis modelados no DRP-MAS. Esses pontos podem ser estendidos pelo usuário do framework e serem definidos da forma considerada mais adequada por cada instância.

- i. **Definir diferentes agentes Mediadores:** A partir do framework podem ser definidos diferentes tipos de Mediadores, isto é, mediadores que criam agentes Diagnóstico e de Recomendação com diferentes estratégias. Para que isso seja possível, basta estender a classe *PlanCreateAgents* (usa o padrão Template Method) e implementar o método *createAgents* ilustrado na **Figura 13**.
- ii. **Estratégia para definir o processo de diagnóstico:** Para definir a estratégia a ser usada para determinar o diagnóstico, a instância deve implementar o método *performingDiagnosis* da classe *PlanDiagnosing* (usa o padrão Template Method) ilustrada na **Figura 14**.
- iii. **Algoritmos de inteligência artificial:** Para realizar o processo de diagnóstico diferentes algoritmos de Inteligência Artificial podem ser usados. O DRP-MAS oferece três algoritmos: forward chaining, backward chaining e lógica fuzzy. Para utilizá-los basta implementar os seguintes métodos: *init* e *importVariables* (ver subseção 4.2.4). Para o algoritmo forward chaining é oferecido um serviço extra que permite realizar inferência de diagnóstico a partir das informações não fornecidas por um agente Solicitante. Esse serviço é oferecido pelo método *getDiagnosesRelated* e é usado pelo método *execute* da classe *ForwardChaining* ilustrada na **Figura 16**.
- iv. **Estratégias para definir o processo de recomendação:** Como mencionado na subseção 4.2.5, o processo de recomendação possui três passos. Para cada um desses passos uma estratégia deve ser definida pela instância do framework. O primeiro passo deve usar uma estratégia de seleção de planos que é representado a partir do método *selectingPlans* da classe *PlanSelectingPlans*. Já o segundo passo usa

uma estratégia para verificar quais planos precisam de agentes, além de solicitar agentes candidatos para o agente de Reputação. Para representar esse passo a instância deve implementar o método *verifyingPlans* da classe *PlanSelectingPlans*. Finalmente, o terceiro passo que deve ter a estratégia para selecionar agentes é realizado a partir do método *verifyRecommendedAgents* da classe *PlanChoosingAgents*.

- v. **Definir diferentes tipos de reputação:** O framework oferece quatro tipos de reputação, sendo uma delas baseada no modelo centralizado (reputação global), e as outras três no modelo descentralizado (Interaction Trust, Witness Reputation e Certified Reputation). Para o modelo descentralizado é possível definir novas reputações a partir da extensão da classe *DecentralisedReputation*.

### 4.5. Diagrama de Caso de Uso

A Figura 26 apresenta o diagrama de casos de uso do framework DRP-MAS. Os detalhes de cada caso de uso são apresentados em seguida.

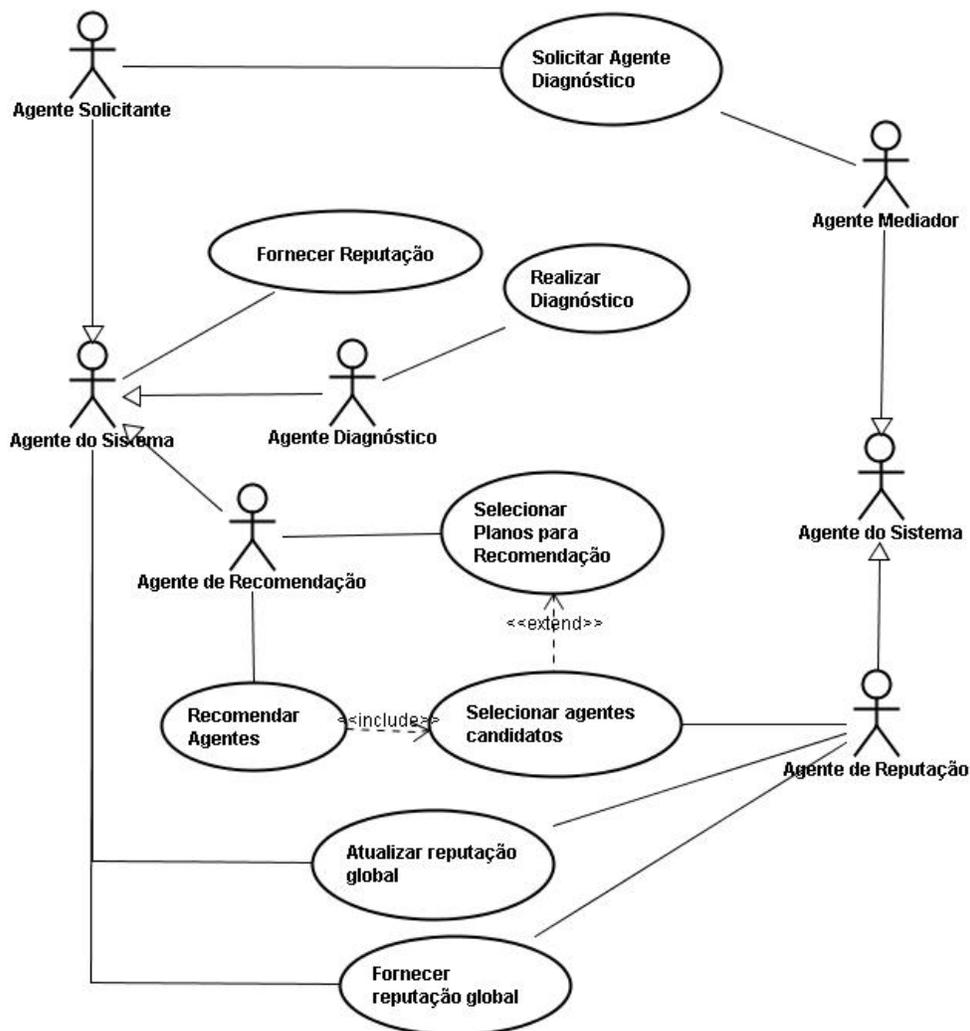


Figura 26. Diagrama de caso de uso do DRP-MAS

#### 4.5.1. Fornecer Reputação

**Ator primário:** Agente do Sistema

**Objetivo:** O objetivo desse caso de uso é fornecer a reputação calculada por um agente do sistema para algum outro. As reputações que podem ser calculadas são aquelas representadas no modelo descentralizado. O agente responsável pela execução desta ação é o agente do sistema.

**Pré-condições:** Nenhuma

**Fluxo Normal:**

1. Este caso de uso é iniciado quando um agente envia uma mensagem para outro agente requisitando uma reputação calculada.
2. O agente que recebeu a requisição consulta a base de reputações para pegar a reputação solicitada.
3. Ao encontrar a reputação solicitada, o agente envia a informação para o agente que realizou a requisição.
4. O agente que realizou a solicitação recebe a informação e a utiliza da forma que desejar.

**Fluxo Alternativo:** Quando o agente não possui reputação

- A.2. O agente que recebeu a solicitação faz a pesquisa em sua base de reputações. Caso não encontre a reputação solicitada, envia nulo para o agente que realizou a requisição.

**Atores Secundários:** Nenhum

#### 4.5.2. Solicitar Agente Diagnóstico

**Ator primário:** Agente Solicitante

**Objetivo:** O objetivo deste caso de uso é representar a situação em que um agente Solicitante falha em alcançar um objetivo desejado. Visando alcançá-lo, o agente procura requisitar recomendações de execução a partir de uma solicitação de um agente Diagnóstico para um agente Mediador, responsável por criar tanto um agente Diagnóstico como um agente de Recomendação.

**Pré-condições:** Nenhuma

**Fluxo Normal:**

1. Um agente Solicitante realiza a execução de algum plano para alcançar um objetivo desejado. Ao término da execução, ele percebe que o objetivo não foi alcançado. Para que seja possível alcançá-lo, o agente solicita um agente Diagnóstico para um agente Mediador.
2. Quando o agente Mediador recebe a solicitação, ele cria um agente de Recomendação e um agente Diagnóstico exclusivos para o agente Solicitante. Nesse processo de criação, o agente Diagnóstico recebe o nome do agente de Recomendação, enquanto o agente de Recomendação recebe o nome do agente Solicitante, para que a comunicação entre os agentes possa ser realizada.
3. Após a criação do agente Diagnóstico e o agente de Recomendação, o Mediador informa o nome do agente Diagnóstico ao agente Solicitante.
4. O agente Solicitante recebe o nome do seu agente Diagnóstico.

**Fluxo Alternativo:** Nenhum

**Atores Secundários:** Agente Mediador, Agente Diagnóstico e Agente de Recomendação.

### 4.5.3. Realizar Diagnóstico

**Ator primário:** Agente Diagnóstico

**Objetivo:** O objetivo deste caso de uso é descrever o processo de diagnóstico realizado por um agente Diagnóstico.

**Pré-condições:** O agente Solicitante deve ter recebido a partir do agente Mediador o nome do seu agente Diagnóstico.

**Fluxo Normal:**

1. O agente Solicitante envia ao agente Diagnóstico um conjunto de dados sobre a execução do plano que não conseguiu alcançar o objetivo desejado.
2. A partir das informações fornecidas pelo agente Solicitante, o agente Diagnóstico encontra o diagnóstico que descreve a razão que impediu o agente Solicitante alcançar o objetivo.
3. Agente Diagnóstico informa ao agente de Recomendação o diagnóstico encontrado e o conjunto de dados enviado pelo agente Solicitante.

**Fluxo Alternativo:** Diagnóstico não encontrado.

A.2. Caso o agente Diagnóstico não encontre o diagnóstico da execução realizada pelo agente Solicitante, uma mensagem é enviada ao agente de Recomendação informando o ocorrido.

**Atores Secundários:** Agente de Recomendação e agente Solicitante.

### 4.5.4. Selecionar Planos para Recomendação

**Ator primário:** Agente de Recomendação.

**Objetivo:** O objetivo deste caso de uso é que o agente de Recomendação selecione planos alternativos que permitam o agente Solicitante alcançar o objetivo desejado.

**Pré-condições:** Agente de Recomendação receber o diagnóstico encontrado pelo agente Diagnóstico.

**Fluxo Normal:**

1. Agente de Recomendação seleciona planos para recomendação a partir do conjunto de dados provido pelo agente Solicitante (ex: objetivo desejado) e a partir do diagnóstico encontrado.
2. Após a seleção dos planos, o agente de Recomendação verifica se algum dos planos precisa de serviços oferecidos por outros agentes de software. Caso não seja necessário, as recomendações são enviadas para o agente Solicitante.
3. Agente Solicitante recebe as recomendações e seleciona aquela que desejar.

**Fluxo Alternativo:** Planos selecionados precisam de serviços oferecidos por agentes da aplicação.

A.3. Quando o agente de Recomendação percebe que alguns dos planos selecionados precisam de serviços oferecidos por agentes, para que suas execuções sejam finalizadas com sucesso, uma mensagem é enviada ao agente de Reputação solicitando agentes que possam ser usados pelos planos (caso de uso “Selecionar agentes candidatos”).

**Atores Secundários:** Agente Solicitante e Agente Reputação.

#### 4.5.5. Selecionar Agentes Candidatos

**Ator primário:** Agente Reputação

**Objetivo:** O objetivo deste caso de uso é permitir que o agente de Reputação possa selecionar agentes baseando-se em suas reputações, para que em seguida informe os escolhidos ao agente de Recomendação.

**Pré-condições:** Agente de Recomendação solicita agentes candidatos.

**Fluxo Normal:**

1. Agente de Reputação recebe uma requisição do agente de Recomendação para que sejam selecionados agentes com boas reputações e que forneçam algum dos serviços solicitados. O agente de Reputação usa para a seleção dos agentes o perfil do agente Solicitante, encaminhado pelo agente de Recomendação, responsável por definir a mínima reputação aceitável para negociar com os agentes.
2. Agente de Reputação seleciona agentes candidatos e informa ao agente de Recomendação.

**Fluxo Alternativo:** Nenhum agente candidato foi encontrado.

A.2. Quando o agente de Reputação não encontra agentes candidatos, ele informa uma lista de agentes vazia.

**Atores Secundários:** Agente de Recomendação.

#### 4.5.6. Recomendar Agentes

**Ator primário:** Agente de Recomendação

**Objetivo:** Agente de Recomendação relaciona quais agentes devem ser recomendados para cada plano escolhido, e em seguida informa ao agente Solicitante as recomendações selecionadas.

**Pré-condições:** Agente de Reputação informa os agentes candidatos para o agente de Recomendação.

**Fluxo Normal:**

1. Agente de Recomendação realiza o filtro dos agentes candidatos a partir do conjunto de dados informado pelo agente Solicitante.
2. Agente de Recomendação relaciona quais agentes poderão ser usados em quais planos selecionados.
3. Agente de Recomendação fornece suas recomendações para o agente Solicitante.

**Fluxo Alternativo:** Plano sem agente para recomendação.

A.2. Quando o agente de Recomendação não encontra agentes que possam ser recomendados para algum plano, apenas aqueles planos que não precisam de agentes ou aqueles que já possuem agentes é que são recomendados e informados para o agente Solicitante.

**Atores Secundários:** Agente de Reputação e agente Solicitante.

#### 4.5.7.

#### Atualizar Reputação Global

**Ator primário:** Agente Reputação

**Objetivo:** O objetivo deste caso de uso é atualizar a reputação global de algum agente do sistema. O Agente de Reputação é responsável por mediar essa atualização, já que somente ele pode ter acesso direto à base centralizada composta pelas reputações globais.

**Pré-condições:** Nenhuma.

**Fluxo Normal:**

1. Um agente do Sistema solicita a atualização da reputação global de algum agente. Assim, uma mensagem é enviada para o agente de Reputação, contendo uma informação que será usada como base para a atualização.
2. O agente de Reputação recebe a mensagem enviada pelo agente do sistema. Em seguida realiza a atualização solicitada.

**Fluxo Alternativo:** Nenhum.

**Atores Secundários:** Agente do Sistema.

#### **4.5.8. Fornecer Reputação Global**

**Ator primário:** Agente de Reputação

**Objetivo:** O objetivo deste caso de uso é fornecer a reputação global de algum agente a partir de uma solicitação realizada por um agente do Sistema.

**Pré-condições:** Nenhuma.

**Fluxo Normal:**

1. Agente do Sistema solicita a reputação global de um agente. Para isso, uma mensagem é enviada ao agente de Reputação.
2. Agente de Reputação recebe a mensagem enviada pelo agente do Sistema. A partir disso, ele acessa a base centralizada composto por reputações globais e realiza a busca pela reputação solicitada.
3. Ao encontrar, o agente de Reputação envia a reputação global para o agente do Sistema.

**Fluxo Alternativo:** Reputação global não encontrada

A.3 Quando a reputação global não é encontrada, o agente de Reputação informa nulo para o agente do Sistema.

**Atores Secundários:** Agente do Sistema.

## 4.6. Diagramas de Seqüência

Nesta seção apresentamos os diagramas de seqüência referente aos principais casos de uso: Solicitar Agente Diagnóstico, Realizar Diagnóstico, Selecionar Planos para Recomendação, Selecionar Agentes Candidatos e Recomendar Agentes.

### 4.6.1. Solicitar Agente Diagnóstico

A **Figura 27** apresenta um diagrama de seqüência que ilustra o cenário em que um agente solicita um agente Diagnóstico a um Mediador. A partir da solicitação dois agentes são criados: Diagnóstico e Recomendação.

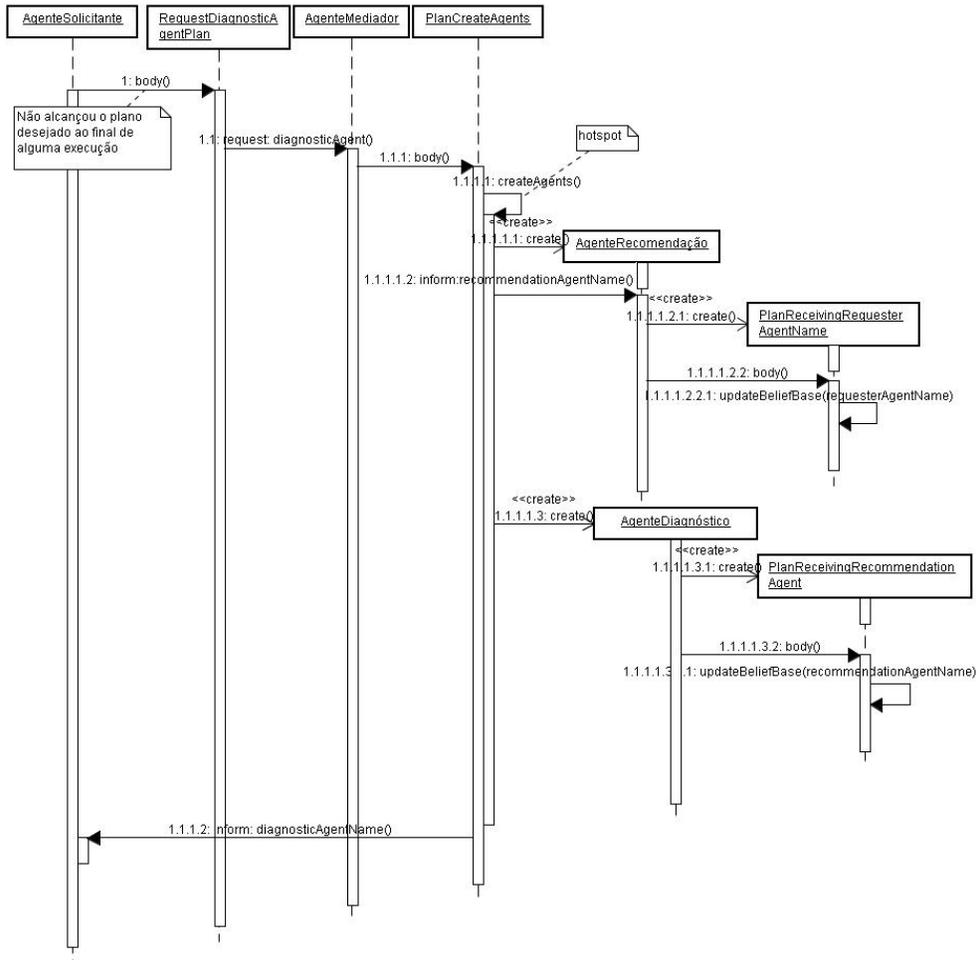


Figura 27. Diagrama de seqüência – Solicitar Agente Diagnóstico.

### 4.6.2. Realizar Diagnóstico

A **Figura 28** mostra um diagrama de seqüência que ilustra o processo base para realizar diagnósticos. Note que nesse exemplo consideramos que foi encontrado um diagnóstico real (MainDiagnosis), a partir das informações fornecidas pelo agente Solicitante, e um diagnóstico inferido (InferredDiagnosis).

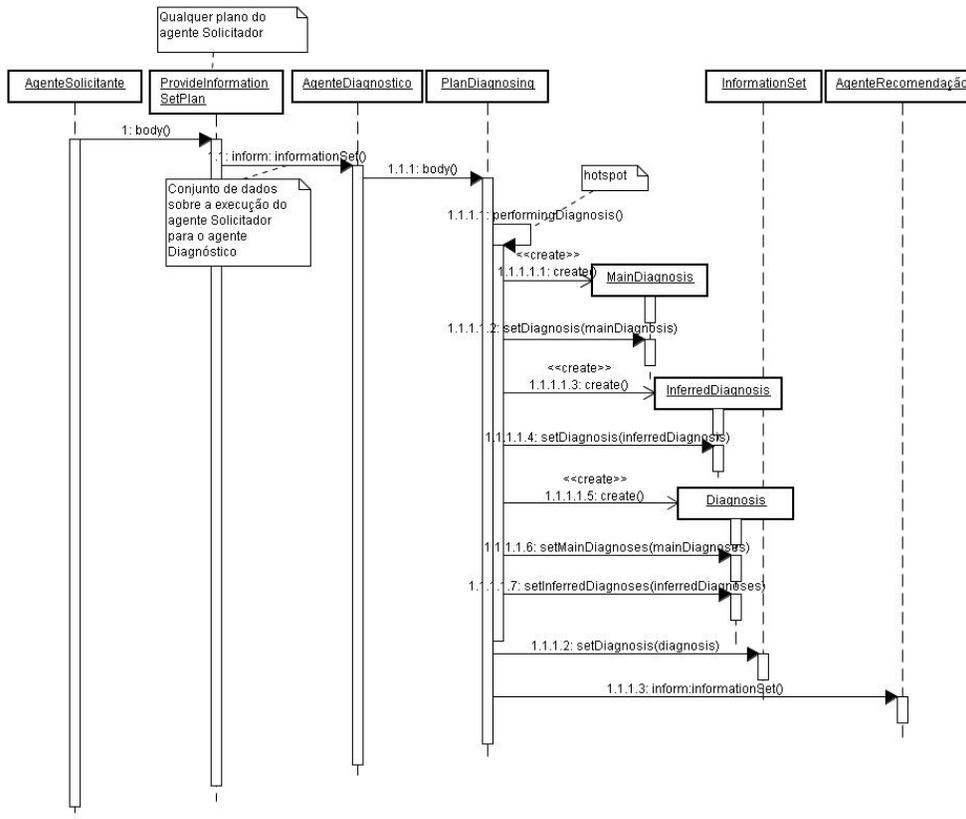


Figura 28. Diagrama de seqüência – Realizar Diagnóstico.

### 4.6.3. Selecionar Planos para Recomendação

O diagrama da **Figura 29** apresenta a parte do processo de recomendação em que os planos a serem recomendados são selecionados, além da verificação que é realizada para determinar quais planos precisam de serviços oferecidos por outros agentes do sistema. Detalhes descritos na subseção 4.2.5.

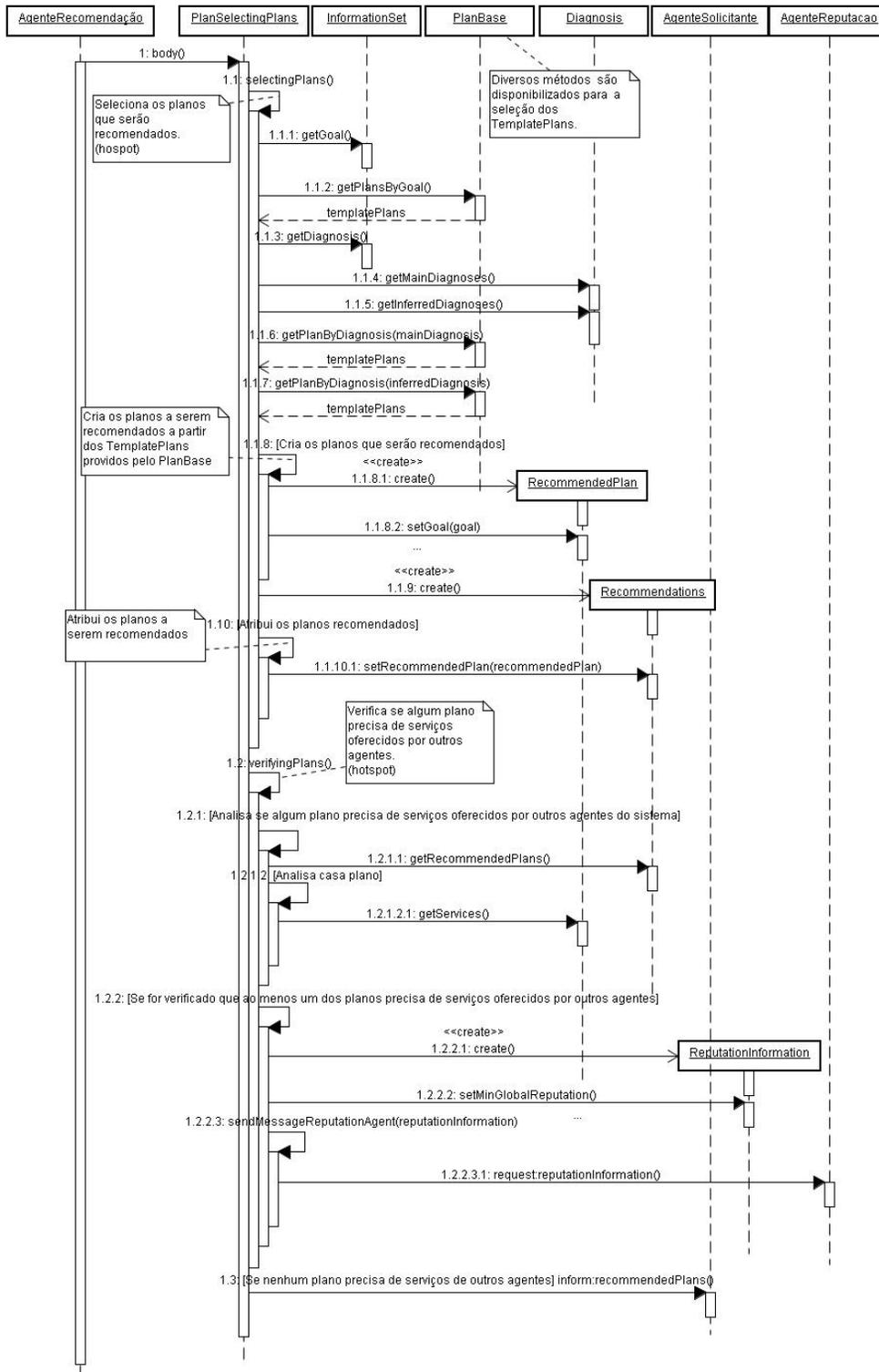


Figura 29. Diagrama de seqüência – Selecionar planos para Recomendação.

#### 4.6.4. Selecionar Agentes Candidatos

Quando um agente de Recomendação solicita agentes que tenham boa reputação, o agente de Reputação é acionado. O diagrama da **Figura 30** ilustra os métodos oferecidos pelo framework para que o agente de Reputação possa realizar a seleção de agentes candidatos.

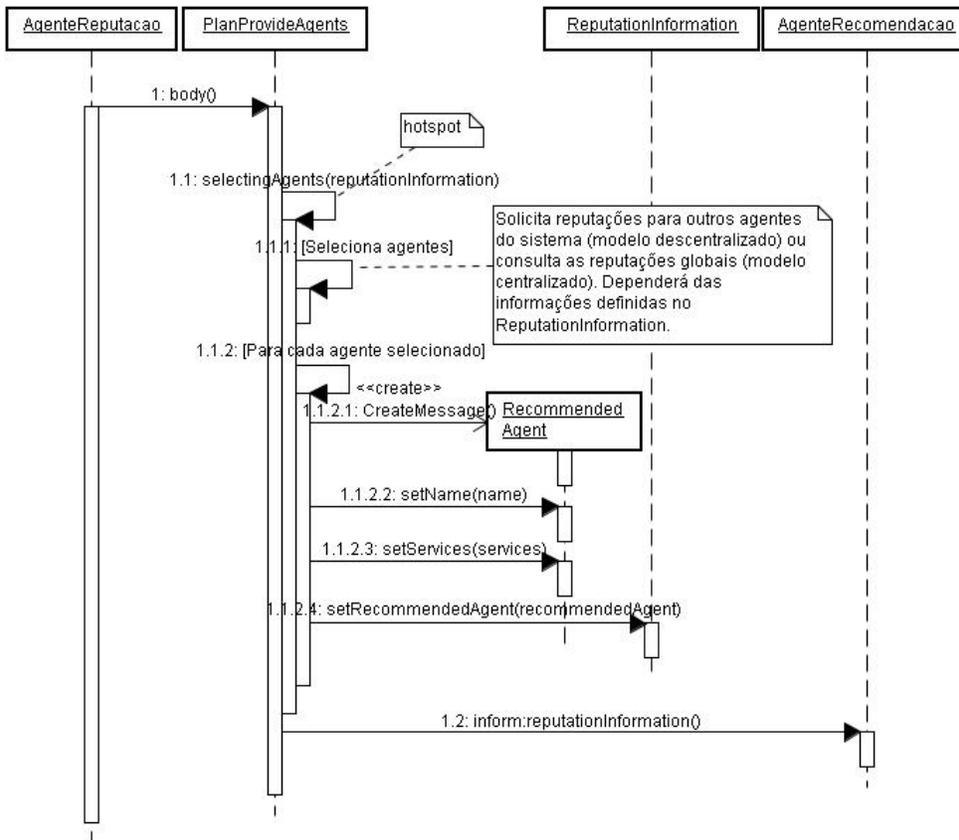


Figura 30. Diagrama de seqüência – Selecionar Agentes Candidatos.

#### 4.6.5. Recomendar Agentes

A **Figura 31** apresenta o cenário em que o agente de Recomendação seleciona os agentes que serão realmente recomendados para cada plano escolhido.

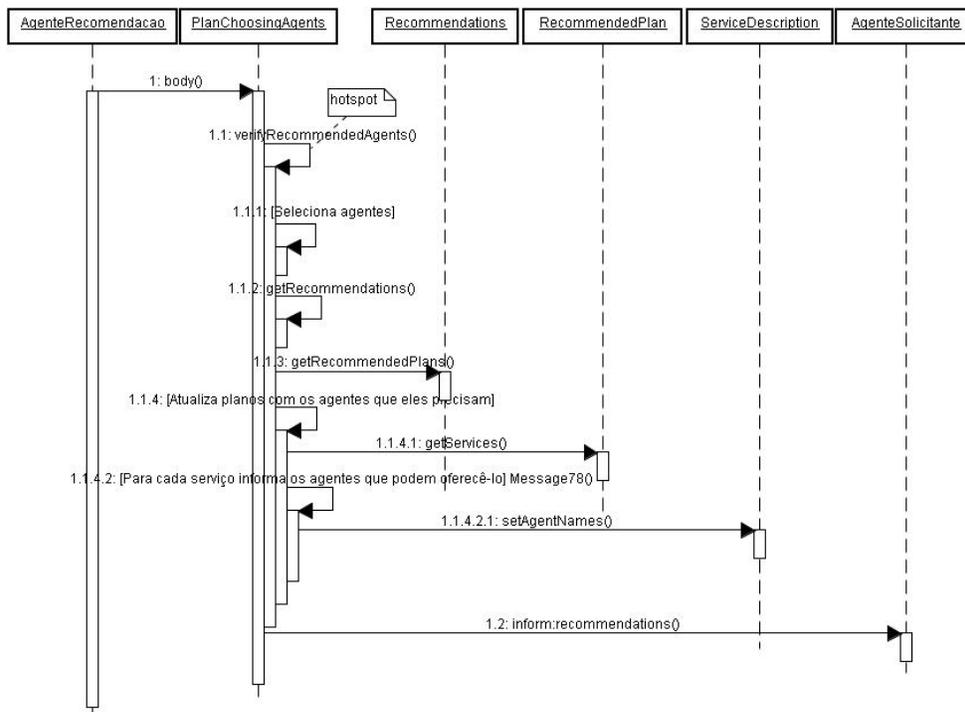


Figura 31. Diagrama de seqüência – Recomendar Agentes.

#### 4.7. Outra Abordagem para o DRPMAS

Com o objetivo de testar nossa abordagem para gerar diagnósticos e recomendações, decidimos implementar os conceitos presentes na implementação JADEX em outra plataforma para sistemas multi-agentes. Nova abordagem foi então implementada utilizando o Agent Society Framework (ver subseção 2.3), framework que representa o conceito sociedade de agentes.

O framework ASF aplica diversos conceitos que não estão presentes no JADEX, como, por exemplo, organização, papel e ambiente. Ao criar o DRPMAS a partir desse framework, conseguimos comprovar que mesmo em diferentes frameworks de sistemas multi-agentes com diferentes conceitos, a proposta apresentada pode ser aplicada.

Nesta seção damos enfoque à implementação dos três principais pontos da nossa abordagem: (i) o processo para diagnosticar o problema, (ii) o processo para prover recomendações e (iii) o módulo de reputação. Esses pontos foram escolhidos porque foram aqueles que sofreram mais modificações na adaptação

do DRP-MAS ao framework ASF, dado que vários conceitos presentes no ASF não estavam presentes na primeira implementação do DRP-MAS

#### **4.7.1.1. Processos de Diagnosticar e Recomendar**

Assim como mencionado na subseção 2.3, o ASF representa o conceito *sociedade de agentes*, isto é, agentes desempenham papéis disponibilizados por organizações. O conceito de papel de agentes pode ser de grande importância ao realizar diagnósticos e prover recomendações. Um diferente diagnóstico pode ser produzido dependendo do papel desempenhado pelos agentes e diferentes planos alternativos podem ser sugeridos dependo do papel desempenhado pelo agente Solicitante e pelos agentes com os quais interagiu.

Outro conceito importante, definido pelo framework REPORT e utilizado nesta segunda implementação do DRP-MAs, é *norma*. Utilizando-se normas pode-se definir as obrigações, permissões ou proibições de um agente. Como consequência, é possível identificar as violações dos agentes à estas normas. Dessa forma, o conceito norma também pode ser útil tanto para o processo de diagnóstico como para o de recomendação.

Em resumo, além das informações mencionadas na seção 4.2.2, os seguintes dados também podem ser usados na nova abordagem:

1. Papel: Tanto o papel desempenhado pelo agente Solicitante como os papéis desempenhados pelos agentes que proveram serviços durante a execução do plano que falhou podem ser úteis para realizar diagnósticos e fornecer recomendações.
2. Norma: A violação de uma norma pode ser a responsável pelo agente Solicitante não alcançar o objetivo desejado. Dessa forma, essa informação pode ser de grande utilidade para encontrar diagnósticos. Já em relação ao processo de recomendação, aqueles planos que não estejam associados às normas violadas podem ser recomendados.

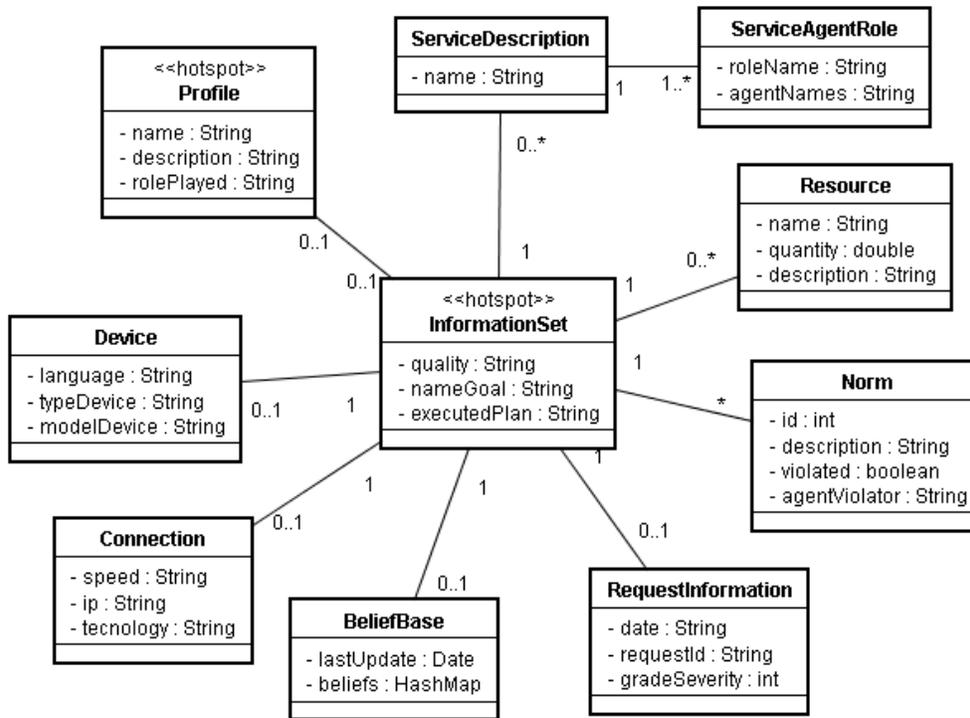


Figura 32. Diagrama de classes DRPMAS com ASF e REPORT.

Do conjunto de dados fornecido pelo agente Solicitante para o agente Diagnóstico apresentado na subseção 4.2.2, pode-se perceber que nessa nova abordagem (**Figura 32**) três mudanças foram realizadas. A primeira mudança é ocorreu na classe Profile, que agora indica qual papel estava sendo desempenhado pelo agente Solicitante quando não conseguiu alcançar o objetivo desejado. A segunda mudança refere-se à associação da classe InformationSet com a classe Norm. E a terceira mudança encontra-se na classe ServiceDescription, que agora está associada com instâncias da classe ServiceAgentRole, responsável por identificar quais foram os papéis desempenhados pelos agentes que forneceram serviços para o Solicitante durante a execução do seu plano.

Tanto o processo de diagnóstico como o processo de recomendação passam a utilizar as novas informações representadas. Perceba que todas as definições presentes no DRP-MAS e apresentadas nas subseções anteriores continuam presentes e sem mudanças.

#### 4.7.1.2. Módulo de Reputação

Nesta subseção é descrito como o framework REPORT foi utilizado pela nova implementação do DRP-MAS, e quais reputações podem ser usadas. Assim como mencionado na subseção 2.4.1, o *framework* REPORT implementa um mecanismo de reputação baseado em testemunhos em violações de normas. Os sistemas de reputação instanciados a partir do *framework* REPORT deverão ser implementados nas organizações que compõem a aplicação. As organizações serão então capazes de avaliar e armazenar as reputações dos agentes com base nos testemunhos sobre violações de normas. As normas e os papéis que os agentes irão desempenhar são definidos pelas organizações da aplicação. Cada sistema de reputação deverá ser capaz de receber os veredictos dos testemunhos, calcular e manter as reputações dos agentes atualizadas, e disponibilizar a reputação para todos os agentes do sistema multi-agente.

As reputações utilizadas são as seguintes: global, norma, papel e local. A primeira reputação está relacionada ao conceito reputação global já mencionado na seção 4.2.6. A segunda refere-se à reputação do agente por norma violada, enquanto que a terceira se relaciona ao papel desempenhado pelo agente. Finalmente a quarta reputação, local, considera todas as violações de normas e falsos testemunhos de um agente dentro de uma organização.

O framework REPORT define três agentes: Attendant, Evaluator e Auxiliary. O primeiro agente é responsável pelo recebimento de requisições dos agentes de software e/ou do subsistema de julgamento (ver subseção 2.4.1), que solicitam a reputação de um determinado agente. O agente Attendant também fica responsável pelo recebimento de veredictos enviados pelo subsistema de julgamento sobre testemunhos de normas violadas. Ao receber um veredicto o agente Attendant cria um agente Evaluator. O agente Evaluator é responsável por receber o veredicto do agente Attendant, classificar-lo como uma violação de norma ou falso testemunho e então alterar a reputação do agente considerado culpado. E finalmente o agente Auxiliary é o responsável pela atualização de todas as reputações de todos os agentes de tempos em tempos.

Para integrar o framework REPORT com o agente de Reputação do DRP-MAS, explicado na seção 4.2.6, este agente troca mensagens com o agente Attendant. A **Figura 33** ilustra os tipos de interações que podem ser realizadas

entre o agente de Reputação e o Attendant. Percebe-se que quando o agente de Reputação deseja uma ou mais reputações, ou quando deseja atualizar alguma reputação, o agente envia uma mensagem para o agente Attendant.

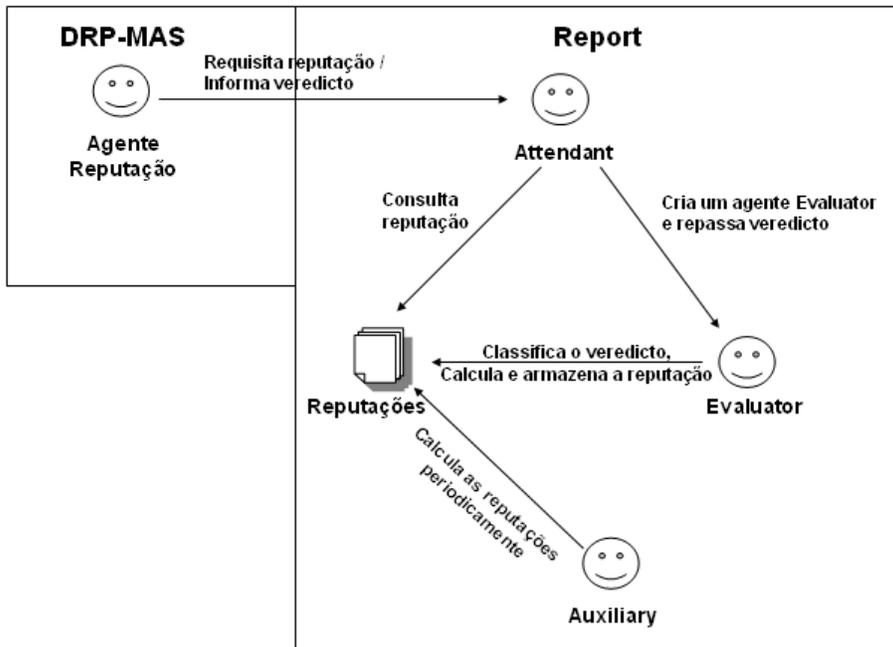


Figura 33. Relação do agente Reputação com agentes do framework REPORT.

Para que uma reputação seja alterada, o agente de Reputação deverá prover um veredicto. Esse veredicto informa se o agente é culpado ou inocente em relação a alguma norma. Assim, quando um agente do sistema deseja alterar alguma reputação global, ele pode encaminhar o veredicto para o agente de Reputação, que funciona como um mediador com o agente Attendant. Além disso, o agente de Reputação segue a mesma idéia de prover agentes candidatos a partir de uma solicitação de um agente de Recomendação (ver subseção 4.2.5).

Com essa nova abordagem, podemos concluir que foi aplicado um modelo de reputação centralizado, diferente daqueles aplicados na abordagem original do DRP-MAS. Mesmo com diferentes abordagens, o conceito proposto pelo DRP-MAS para realizar diagnóstico e recomendação baseando-se em reputações pode ser aplicado a partir de diferentes tecnologias (ex: frameworks.) e definições (ex: modelos de reputação).

#### 4.8. Desafios Tratados no DRP-MAS

Nesta seção são apresentados como os desafios, mencionados na subseção 2.6, foram tratados pelo framework DRP-MAS. Tais explicações são realizadas brevemente, já que nas seções anteriores o framework foi apresentado em detalhes. A seguir, os desafios com seus correspondentes tratamentos.

1. **Decidindo como analisar o comportamento dos agentes:** O DRP-MAS considera que cada agente deve ser capaz de levantar um conjunto de dados sobre sua execução. Esses dados devem ser informados para um agente diagnóstico, responsável por realizar o processo de diagnóstico, que em seguida encaminha o diagnóstico encontrado para um agente de Recomendação. Este agente é aquele que envia as recomendações encontradas para o agente solicitante.
2. **Selecionando dados para diagnósticos:** Este desafio está relacionado à seleção de dados apropriados para encontrar um diagnóstico que esteja ligado à falha da execução realizada pelo agente. Para encontrar um diagnóstico, diferentes informações foram definidas pelo framework, como, por exemplo: objetivo desejado, plano executado, recursos utilizados, perfil do agente, além de serviços solicitados para outros agentes durante a execução do plano. A listagem dessas informações é apresentada na seção 4.2.2.
3. **Determinando estratégias para diagnósticos:** Para permitir que uma abordagem flexível pudesse ser adaptada para diferentes domínios, o framework DRP-MAS definiu um plano (PlanDiagnosing), que permite realizar diferentes tipos de diagnóstico, como um hotspot do sistema. Para auxiliar na criação de estratégias, o framework também oferece algoritmos de inteligência artificial, mencionadas na seção 4.2.4.

4. **Determinando estratégias para recomendações:** Para permitir que recomendações possam ser representadas a partir do DRP-MAS, um processo composto por três etapas foi definido: (i) selecionando planos, (ii) verificando planos selecionados e (iii) selecionado agentes (ver subseção 4.2.5). Além disso, os algoritmos de inteligência artificial oferecidos pelo framework também podem ser usados pelo processo de recomendação, assim como no processo de diagnóstico.
  
5. **Determinando agentes confiáveis:** Como mencionado na seção 2.6, durante a execução de um plano, um agente pode precisar interagir com outros agentes. O serviço provido por este agente ou a informação enviada pode determinar se o objetivo do agente, que solicitou o serviço ou pediu a informação, será ou não alcançado. Assim, o DRP-MAS permite definir se algum agente deve ou não ser mantido na lista de possíveis agentes para futuras interações quando um diagnóstico é formulado e verifica-se que ele foi o responsável pela execução não bem sucedida de um plano. O framework se baseia no uso de reputações associadas à agentes para escolher os futuros parceiros. Para representar o conceito reputação um módulo de reputação foi criado (ver seção 4.2.6). A partir dele, diferentes tipos de reputação podem ser definidos, além daqueles oferecidos pelo framework.
  
6. **Provendo suporte para sistemas de computação ubíqua:** Para oferecer suporte para sistemas de computação ubíqua que desejam realizar diagnósticos e recomendações, o framework permite que o agente solicitante informe ao agente diagnóstico o dispositivo (ex: celular, PDA, etc) e o tipo de conexão (velocidade, tecnologia e endereço IP) utilizados durante alguma execução (detalhes na seção 4.3). Caso sejam necessários mais dados relacionados à computação ubíqua para realizar diagnósticos e recomendações, novas informações podem ser representadas. Essa é uma das vantagens do uso do framework, já que esse conjunto de dados foi definido como um ponto flexível,

7. **Representando perfis de agentes:** Como cada agente pode possuir características e propriedades particulares, o DRP-MAS define o conceito perfil como um ponto flexível (classe Profile) A partir disso, diferentes perfis podem ser representados para uma ou mais instâncias do framework.