2
Conceitos Importantes

Nesta seção são apresentados conceitos, que serviram de motivação para o trabalho proposto e que são referenciados no decorrer desta documentação. Inicialmente é apresentado o framework JADEX [Poukahr and Braubach, 2007a] [Poukahr and Braubach, 2007b] [Poukahr and Braubach, 2007c], responsável por representar o conceito de agentes de software orientados a objetivos. A seguir, a plataforma JADE LEAP [Caire, 2003] usada para criar aplicações móveis (ex: PDAs e celulares) e o framework ASF [Silva et al., 2004] [Costa, 2006], responsável por representar o conceito sociedade de agentes, são explicados. Para finalizar, conceitos e trabalhos voltados à área de reputação e computação ubíqua são apresentados.

2.1.
JADEX

JADEX é um framework baseado no conceito de sistemas multi-agentes que estende Jade [Bellifemine et al., 2007a] [Bellifemine et al., 2007b] permite a criação de agentes orientados a objetivos seguindo o modelo Belief-Desire-Intention (BDI) [Braubach et al., 2003]. Tais agentes são criados a partir das linguagens Java e XML.

Na **Figura 1** é apresentada a arquitetura abstrata do framework JADEX. Pode-se perceber que um agente é considerado uma caixa preta, que recebe e envia mensagens. Todos os tipos de eventos, como por exemplo, as mensagens recebidas ou os objetivos servem como entrada (input) para a reação do mecanismo de deliberação (reaction deliberation mechanism), que encaminha os eventos para planos selecionados a partir da biblioteca de planos oferecida. No JADEX, a reação do mecanismo de deliberação é um componente do agente, enquanto que os outros componentes são agrupados em módulos reusáveis denominados de capacidades (capabilities). A seguir, tais componentes são explicados em detalhes.

PUC-Rio - Certificação Digital Nº 0621276/CA

1. Crenças (Beliefs): Para o JADEX as crenças (beliefs) correspondem a sua base de conhecimento, onde os objetos podem ser armazenados como fatos (cada crença conhecida), ou conjunto de fatos (conjunto de crenças). Essa base de crenças não é uma base puramente passiva para armazenar dados. A base pode ser utilizada para monitorar as condições de estado das crenças.
2. Objetivos (Goals): Os objetivos são os desejos que os agentes desejam alcançar. Para alcançá-los os agentes executam ações definidas em planos.
3. Planos (Plans): Os planos representam os elementos comportamentais de um agente e são compostos por duas partes: cabeça (head) e corpo (body). A especificação da primeira parte refere-se às circunstâncias que levam um plano a ser selecionado, com, por exemplo, o recebimento de um tipo de mensagem e o alcance de um objetivo. Já a segunda parte provê um curso pré-definido das suas ações para alcançar um objetivo.
4. Capacidades (Capabilities): As capacidades representam o agrupamento dos elementos que compõem um agente BDI, como, por exemplo, crenças, objetivos, planos e eventos.

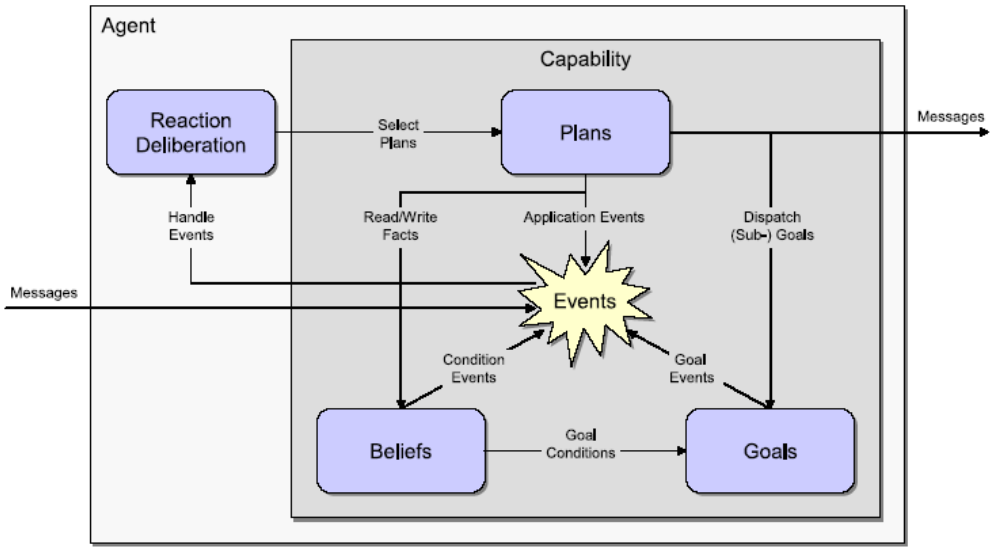

Figura 1. Arquitetura abstrata do JADEX.

Para criar um agente JADEX, torna-se necessário definir dois tipos de componentes (ver Figura 2). O primeiro componente refere-se a um arquivo de definição do agente (Agent Definition File - ADF), responsável por especificar suas crenças, objetivos e planos a serem executados. Por definição, ADF é uma

PUC-Rio - Certificação Digital Nº 0621276/CA

linguagem XML usada para seguir o metamodelo do JADEX BDI, especificado em um XML Schema. Já o segundo componente, referente à parte procedural dos planos (corpo dos planos), é definido utilizando a linguagem de programação Java. Assim, caso um agente deseje executar um plano, este plano deve ser desenvolvido em Java e ser referenciado no arquivo ADF do agente correspondente.

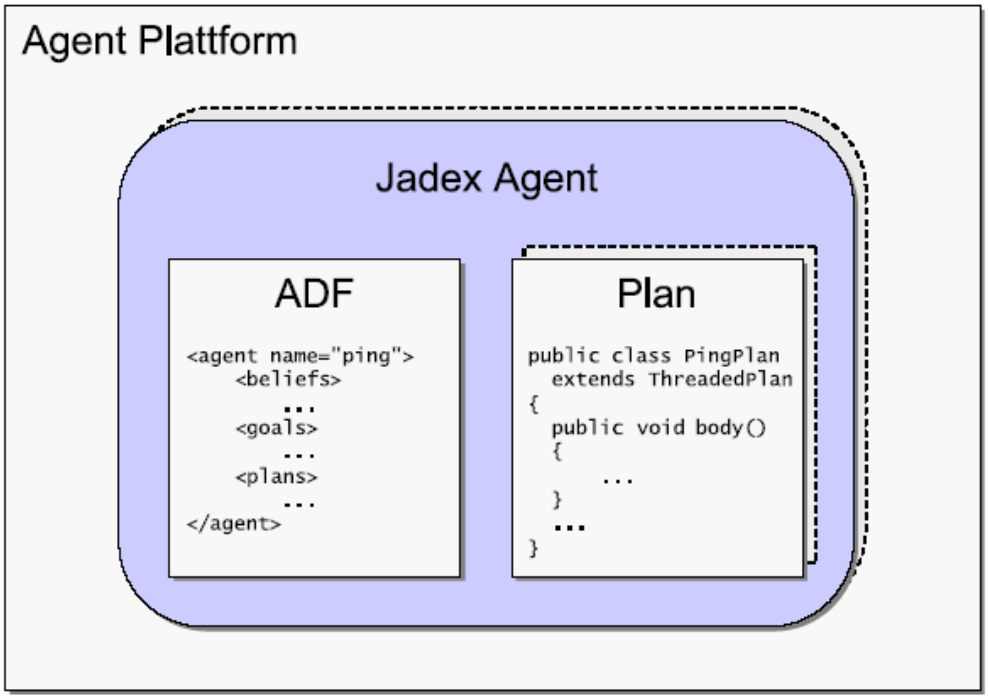

Figura 2. Agente JADEX

JADEX é um framework que permite o uso do framework JADE como um middleware de comunicação. Assim, aplicações desenvolvidas em JADEX e JADE podem ser comunicar sem grandes problemas. Para ajudar o desenvolvimento, o JADEX oferece uma série de ferramentas que permite verificar se um agente foi criado corretamente, quais mensagens eles estão trocando, etc.

2.2. JADE LEAP

JADE-LEAP é uma versão modificada da plataforma JADE, que permite executar agentes FIPA [FIPA, 2008] em diferentes dispositivos, como, por exemplo, celulares que executam Java e dispositivos que executam o framework .Net da Microsoft.

PUC-Rio - Certificação Digital Nº 0621276/CA

Duas plataformas são oferecidas pelo JADE-LEAP: JADE-LEAP PJAVA e JADE-LEAP MIDP. PJAVA oferece suporte para dispositivos portáteis, como PDAs e smart-phones. Já o MIDP executa dispositivos que ofereçam suporte para MIDP 1.0 (ou depois), como, por exemplo, boa parte dos celulares que executam Java.

JADE-LEAP provê um protocolo de comunicação apropriado para pequenos dispositivos. Assim, agentes de software que estão sendo executados em dispositivos móveis são capazes de se comunicar por intermédio de redes sem fio com o agente prestador do serviço, que é executado em um computador fixo. Dessa forma, JADE-LEAP trabalha com altos níveis de comunicação, estabilizando conexões TCP/IP entre diversos contêineres (containers).

A Figura 3 ilustra as plataformas PJAVA e MIDP sendo usadas. Quando o PJAVA é usado, pode-se perceber que um contêiner pode ser definido em um dispositivo, onde algum agente está sendo executado. Esse contêiner pode se conectar com algum contêiner principal (Main-container), localizado em outra estação de trabalho. Já quando o MIDP é usado, o dispositivo utilizado (celular) executa um módulo chamado Front-End, que é definitivamente mais leve do que um contêiner, enquanto que alguma estação de trabalho executa o módulo Back-End, responsável por permitir conexão constante entre eles. Essa máquina, que possui um contêiner, consegue enxergar o contêiner principal localizado em outra máquina.

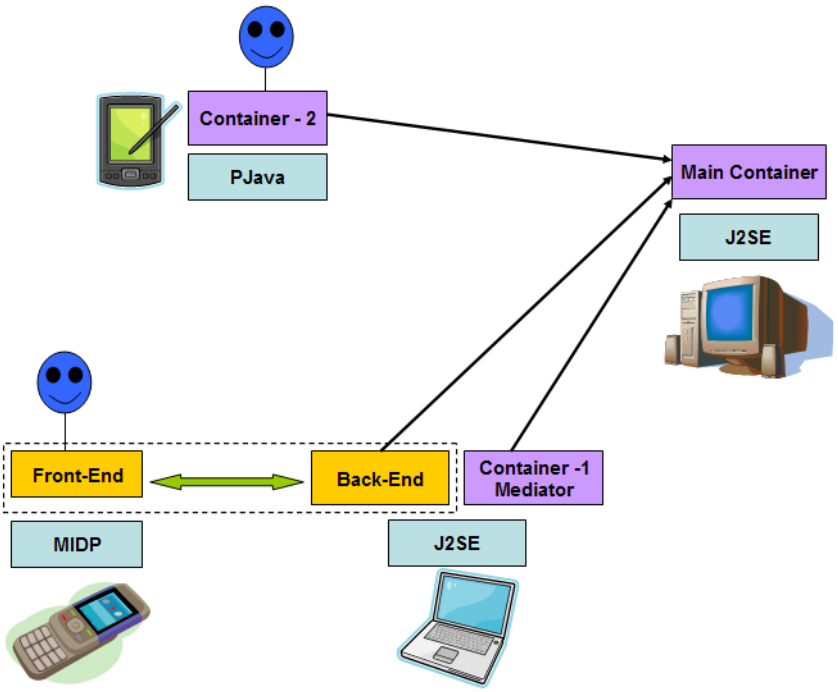

Figura 3. Modos de execução do JADE-LEAP.

PUC-Rio - Certificação Digital Nº 0621276/CA

O desenvolvimento com o JADE-LEAP permite que cenários baseados em computação ubíqua (ver subseção 2.5) possam ser criados facilmente. Devido a isso, essa tecnologia foi escolhida para que a criação de aplicações que seguem o contexto de dispositivos móveis possam ser executadas com agentes de software.

2.3. Agent Society Framework

O Agent Societ Framework (ASF) é um framework FIPA complaint orientado a objetos para implementar sociedades de agentes. Utilizando-o é possível implementar vários agentes desempenhando inúmeros papéis em diferentes organizações. Desta forma o framework lida com agentes, papéis, organizações e ambientes como entidades de primeira ordem. Para cada entidade, um detalhado ciclo de vida é definido descrevendo os estados de execução e eventos que causam as transições de uma entidade a partir de um estado para outro. Baseada em representações gráficas dos modelos de ciclo de vida, os modelos computacionais de cada entidade podem ser explorados. Eles definem os comportamentos das entidades associadas a cada estado descrito nos ciclos de vida.

Na **Figura 4** é apresentada a visão geral de como as principais entidades presentes no ASF são implementadas. Verifica-se a representação de conceitos que já foram citados, como: organização, papel e agente.

Cada agente (Agent) desempenha um ou mais papéis (AgentRole). Quando é definida uma relação desse tipo, o agente poderá ter objetivos (Goal) que serão alcançados por intermédio de planos (Plan) responsáveis por executar um conjunto de ações (Action) que podem ter pré e pós-condições (Condition). Ao executar os planos e ações, os agentes atualizam o conjunto de crenças (Belief) que possuem. Existem dois tipos de objetivos: composto (CompositeGoal) e simples (LeafGoal). O simples nada mais é do que a própria representação de um objetivo qualquer. Já o composto, indica que ele só será alcançado, caso os objetivos que o compõem também sejam. Da mesma maneira, existem dois tipos de crenças: simples e composta. Como no caso dos objetivos, a primeira representa a própria crença, já a outra define que uma delas é composta por várias.

PUC-Rio - Certificação Digital Nº 0621276/CA

Associados aos papéis podem estar objetivos, crenças e ainda ações caracterizadas como obrigações (Duty) ou direitos (Right). Os objetivos de um papel são aqueles que o agente deve tentar atingir enquanto desempenha o papel. Enquanto as crenças de um papel são passadas para o agente assim que este se dispõe a desempenhar o papel. As ações caracterizadas como obrigações são ações que um agente TEM que executar, e as de direito identificam quais ações um agente PODE executar.

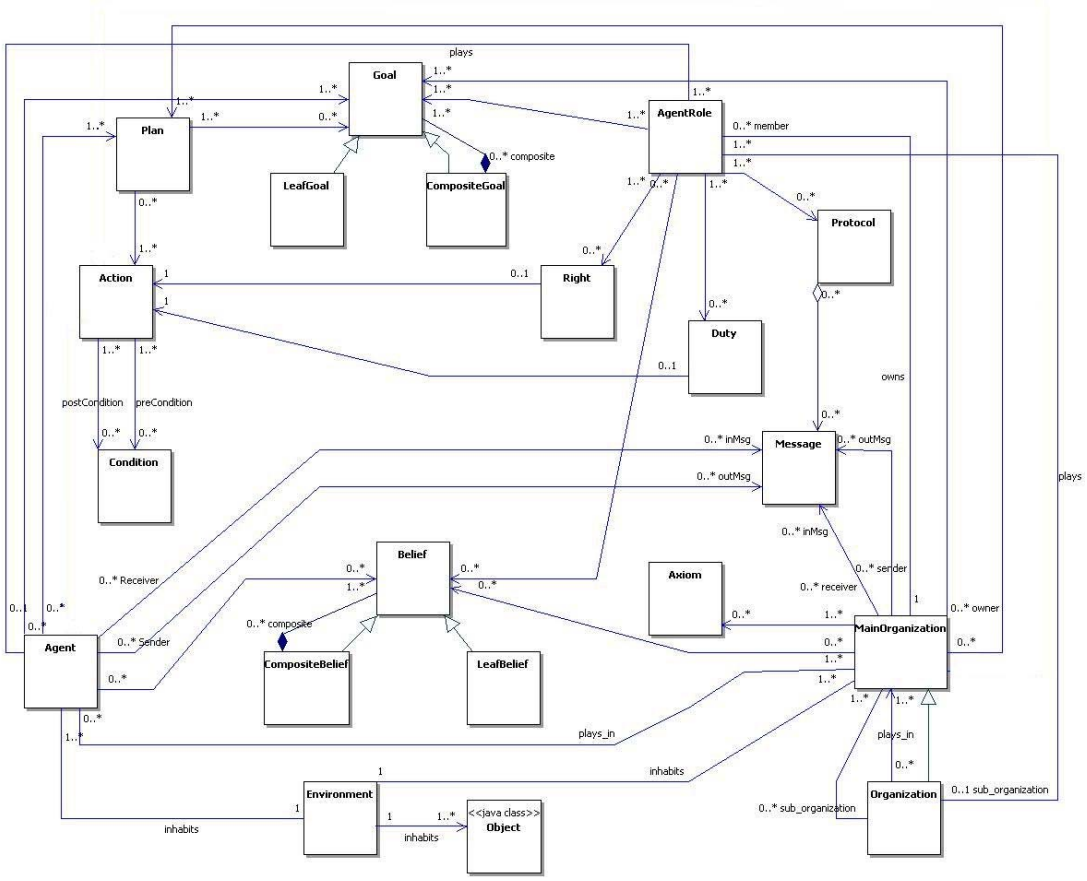

Figura 4. Modelo ASF

Cada papel deverá fazer parte de um contexto de uma organização principal (MainOrganization) ou não (Organization) que poderá determinar axiomas (Axiom) que tanto seus agentes como suas sub-organizações (Organization) deverão respeitar. Além dos agentes poderem desempenhar papéis, uma sub-organização também poderá. Além de definir axiomas, uma organização que atua ativamente em transações da aplicação poderá ter objetivos, planos, ações e crenças.

PUC-Rio - Certificação Digital Nº 0621276/CA

Uma parte bem interessante do framework é a comunicação por mensagens. Para a sua realização, três tipos de relações são possíveis: agentes com organizações, agente com agentes e organizações com organizações. Assim, quando uma dessas é usada, são utilizadas mensagens (Message) que são tratadas por protocolos (Protocol) que são associados com papéis de agentes e/ou de organizações.

Logo, as principais entidades do modelo são aquelas que representam os agentes e as organizações. Elas são hospedadas em algum ambiente (Environment) responsável por guardar objetos (Object) utilizados por elas durante suas transações.

2.4. Reputação

Na literatura existem diversas definições para os conceitos confiança e reputação. Uma dessas definições, apresentada por [Koogan and Houaiss, 1995], considera confiança a segurança ou certeza daquele que tem fé na probidade de alguém, enquanto que reputação refere-se ao conceito atribuído a uma pessoa por parte da sociedade em que vive estritamente ligado ao comportamento e utilizado para medir o grau de confiança.

Voltados para esses conceitos, diversos trabalhos baseados em sistemas multi-agentes foram propostos, como, por exemplo, o Governance Framework [Silva et al, 2007] e o modelo Fire [Huynh et al., 2004]. Ambos serão explicados a seguir em detalhes.

2.4.1. Governance Framework

O Governance framework representa um mecanismo de governança baseado em testemunhos fornecidos por agentes que testemunham fatos e eventos relacionados com normas violadas. O framework define três subsistemas, assim como ilustrado na **Figura 5**. O módulo de julgamento [Duran et al., 2008], Judgement System, é o responsável por receber testemunhos e prover uma decisão ou veredicto para o módulo punição (Punishment System). O sistema pode usar diferentes estratégias para julgar a violação de diferentes normas especificadas pela aplicação. Cada estratégia pode usar a reputação de agentes, calculadas a partir do framework REPORT [Guedes et al., 2008] que é definido na figura como o sub-sistema de reputação (Reputation System), para

PUC-Rio - Certificação Digital Nº 0621276/CA

que assim seja encontrada sua decisão final. O módulo de reputação calcula a reputação dos agentes para que em seguida sejam informadas para o módulo julgamento e para aplicações de agente (Application). As reputações são atualizadas a partir de decisões providas pelo módulo de julgamento sobre as violações de normas. Diferentes normas podem influenciar a reputação de agentes em diferentes momentos. Enquanto que o módulo de punição seleciona as penalidades especificadas em normas para que os agentes considerados culpados possam recebê-las.

O framework REPORT define quatro tipos de reputação: global, papel, norma e local. A primeira reputação representa a média de todas as reputações parciais de algum agente, enquanto que a segunda define a média das reputações parciais de um agente desempenhando um mesmo papel. Já a terceira reputação refere-se à reputação do agente por norma violada, enquanto que a última é aquela que considera todas as violações de normas e falsos testemunhos de um agente dentro de uma organização. Para que essas reputações possam ser calculadas, uma série de informações são usadas, como, por exemplo: poder da norma (grau de importância da norma), controle de vezes que um agente violou alguma norma, a informação se um agente confessou alguma violação feita por ele e o grau de certeza que o sistema possui sobre os dados obtidos para calcular a reputação.

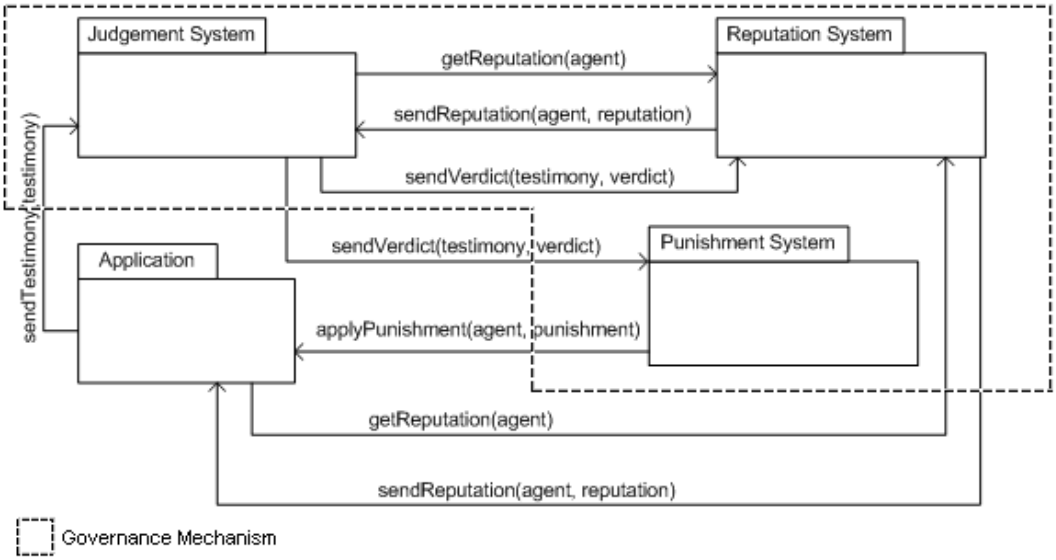

Figura 5. Governance Framework

PUC-Rio - Certificação Digital Nº 0621276/CA

O mecanismo de governança foi implementado usando o framework ASF (ver subseção 2.3), para que pudesse prover suporte para agentes, organizações e papéis. Visando manter tais conceitos, os três sub-sistemas de governança (julgamento, reputação e punição) foram desenvolvidos como organizações separadas responsáveis por interagir com a quarta organização (Application), onde as aplicações de agente estão situadas.

2.4.2. Modelo Fire

O modelo Fire é um dos modelos de reputação mais conhecidos na área de sistemas multi-agentes. Ele integra quatro diferentes tipos de confiança e reputação: interaction trust, role-based trust, witness reputation e certified reputation. O primeiro tipo é referente a experiências passadas a partir de interações diretas, enquanto que o segundo tipo é definido pelas relações baseadas por papel desempenhado pelo agente. Já o terceiro tipo baseia-se em testemunhos fornecidos por agentes, e o quarto a partir de referências providas por outros agentes sobre seu comportamento. Essa abordagem definida pelo modelo Fire torna-se interessante pois permite que um agente possa combinar uma variedade de fontes alternativas de informação (lidar com as incertezas inerentes) e que um agente possa utilizar fontes disponíveis no momento.

Em especial, um novo tipo de reputação chamado de reputação certificada (certified reputation) é proposto. Essa reputação é fundamental quando um agente deseja conhecer a reputação de outro agente, com o qual ele nunca tenha interagido anteriormente, e quando ele não conhece os agentes que já negociaram com o respectivo. Por exemplo, se um agente A nunca interagiu com um agente B, ele não possui nenhuma informação que o permita calcular seu interaction trust. No caso da reputação de testemunho (witness), o agente A não pode não ser capaz de achar testemunhas relevantes sobre o agente B, ou o processo de procura pode ser muito longo. Finalmente, pode não ter ocorrido nenhuma relação baseada em papel com o agente B. Caso todas essas situações aconteçam ao mesmo tempo, o agente A não será capaz de determinar seu grau confiança sobre o agente B. Considerando essas situações, se o agente B pode apresentar as informações certificadas sobre seu desempenho passado para A (na forma de referências a partir de outros agentes que interagiram com ele), o agente A poderá realizar alguma análise sobre a confiança que terá sobre o agente B.

PUC-Rio - Certificação Digital Nº 0621276/CA

2.5. Computação Ubíqua

Segundo [Araujo, 2003], a idéia principal da computação ubíqua é a possibilidade da computação mover-se para fora das estações de trabalho e computadores pessoais (PCs) e tornar-se presente em nossa vida cotidiana. Esse conceito foi apresentado por Marc Weiser [Weiser, 1991] [Weiser, 1993], considerado o pai da computação ubíqua e responsável por vislumbrar no futuro computadores presentes em diversos objetos, como, por exemplo, etiquetas de roupa, canetas, interruptores de luz, etc., de forma invisível para o usuário.

O potencial de aplicações baseadas em computação ubíqua é enorme, considerando o desenvolvimento tecnológico pelo qual o mundo está passando. Situações em que o conceito de computação ubíqua está sendo aplicado seriam aquelas que tratam a conexão, monitoramento e coordenação de dispositivos localizados em casas, edifícios e carros inteligentes através de redes sem fio locais e de longa distância com alta largura de banda. Essas aplicações podem variar desde o controle da luminosidade, temperatura e umidade, até aplicações colaborativas com suporte à mobilidade.

O que tem ajudado bastante na proliferação da computação ubíqua é o avanço tecnológico de diferentes dispositivos, indo desde portáteis, como PDAs, celulares e laptops, até grandes dispositivos compartilhados. Devido esse crescimento, diversos grupos de tecnologia e pesquisa têm investido maior atenção nesse contexto, como, por exemplo, os grupos Mobile Ubiquitous Services & Technologies Group [Mobile, 2008] e The Embedded Systems and Pervasive Computing Lab of the Federal University of Campina Grande [Embedded, 2008]. O primeiro grupo foca sua pesquisa em aplicações de context-aware para pequenos dispositivos, considerando o domínio de saúde, enquanto que o segundo grupo tem como um dos trabalhos principais o desenvolvimento de tecnologias sem fio (wireless) em ambientes industriais. Perceba que a aplicabilidade da computação ubíqua é bem diversificada, sendo limitado somente pela nossa imaginação.

Como essa área apresenta domínios complexos, o trabalho proposto oferece suporte para aplicações que desejam realizar diagnósticos e recomendações. Detalhes sobre tais facilidades são explicados nas próximas seções.

PUC-Rio - Certificação Digital Nº 0621276/CA

2.6. Diagnosticando e Provendo Recomendações

Nesta sub-seção, analisamos os principais desafios encontrados no desenvolvimento do framework DRP-MAS. Estes desafios estão relacionados a realizar diagnósticos e prover recomendações que permitam alcançar um objetivo não alcançado anteriormente por algum agente. Os desafios identificados são apresentados a seguir:

1. **Decidindo como analisar o comportamento dos agentes:** O primeiro desafio foca na análise do comportamento dos agentes. Duas abordagens foram consideradas. Na primeira, a execução de cada agente deveria ser monitorada, no entanto, como essa abordagem viola a privacidade dos agentes, ela foi descartada. Na segunda abordagem, cada agente consegue detectar suas próprias falhas e em seguida prover dados relacionados a estas falhas para serem usados no diagnóstico do problema e na recomendação de novos planos.

2. **Selecionando dados para diagnósticos:** O segundo desafio está relacionado à seleção de dados apropriados para encontrar um diagnóstico que esteja relacionado à falha da execução realizada pelo agente. Para encontrar um diagnóstico, diferentes informações podem ser usadas, como, por exemplo: o registro de falhas e sucessos nas comunicações com outros agentes, além da descrição do problema referente à indisponibilidade de recursos necessários. Uma lista de tais informações foi definida e é apresentada na seção 4.2.2.

3. **Determinando estratégias para diagnósticos:** Como diferentes áreas de aplicação usam informações de domínios específicos para fazer diagnósticos, nosso desafio é definir uma abordagem flexível que permita ser adaptada para diferentes domínios.

PUC-Rio - Certificação Digital Nº 0621276/CA

4. **Determinando estratégias para recomendações:** Assim como as estratégias usadas para prover diagnósticos, as estratégias para gerar recomendações podem ser dependentes do domínio, pois os dados referentes ao domínio possam determinar a recomendação que corresponda ao diagnóstico. Assim, o desafio é oferecer uma abordagem flexível que possa ser adaptada para diferentes domínios, além de uma estratégia padrão (baseada nas informações independentes do domínio) que possa ser usada em diversos domínios.

5. **Determinando agentes confiáveis:** 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, quando um diagnóstico é formulado e verifica-se que um agente foi o responsável pela execução não bem sucedida de um plano, deveria ser viável retirar este agente da lista de possíveis agentes para futuras interações. Para tanto, nosso framework se baseia no uso de reputações associadas a agentes para escolher os futuros parceiros. Segundo [Koogan and Houaiss, 1995], reputação é um conceito atribuído a alguém por parte da sociedade em que vive para medir o grau de confiança e confiança é a segurança ou certeza daquele que tem fé na probidade de alguém.

6. **Provendo suporte para sistemas de computação ubíqua:** Nossa abordagem oferece um componente que realiza diagnósticos e provê recomendações relacionados com sistemas baseados em computação ubíqua. Considerando esse contexto, diversos dados podem ser usados nos processos de diagnóstico e recomendação, como, por exemplo, diferentes dispositivos e tipos de conexões usadas. Os diferentes dispositivos podem afetar uma requisição feita para um agente. Por exemplo, um agente pode ter que enviar um conjunto de dados em um formato específico devido o dispositivo usado na solicitação. O desafio é organizar as limitações envolvendo diferentes dispositivos para entender como pode afetar tanto o processo de diagnóstico como o de recomendação.

PUC-Rio - Certificação Digital Nº 0621276/CA

7. **Representando perfis de agentes:** Cada agente possui características e propriedades particulares que podem ser representadas em um perfil. O desafio é definir um componente que possa ser usado para representar perfis genéricos para agentes de software, além de prover serviços que ajudem as estratégias de diagnóstico e recomendação a utilizar tais perfis.

PUC-Rio - Certificação Digital Nº 0621276/CA