

2

Fundamentação Teórica

Neste capítulo, são apresentados conceitos relevantes para esta dissertação, e que serão mencionados ao longo do trabalho. Inicialmente é dada uma visão geral sobre sistemas multi-agentes (Seção 2.1), que é a tecnologia base para a construção de sistemas auto-organizáveis. Em seguida, é feita uma descrição sobre os principais conceitos de sistemas auto-organizáveis (Seção 2.2), elucidando suas propriedades e caracterizando o fenômeno emergente. Ademais, uma breve explanação sobre o conceito de fenômeno emergente é feita. Os padrões de auto-organização disponibilizados pelo framework proposto neste trabalho de dissertação são apresentados na Seção 2.4. Por fim, o framework Jadex é apresentado na Seção 2.5.

2.1

Sistemas Multi-Agentes

Nos sistemas multi-agentes, a pesquisa está concentrada no estudo, comportamento e construção de uma coleção de agentes autônomos que interagem uns com os outros e com o ambiente [Sycara, 1998]. Um sistema multi-agentes pode ser entendido como uma rede fracamente acoplada de solucionadores de problemas (agentes) que interagem através de um sistema de comunicação para resolverem problemas que estão além da capacidade individual de cada um. Nesta seção, inicialmente, daremos uma visão geral sobre as características dos agentes simples, para em seguida, tratar dos aspectos dos sistemas multi-agentes que focam em problemas de interação e coordenação.

2.1.1

Agente e Ambiente

Wooldridge [Wooldridge, 2002] define um agente como um sistema computacional que está situado em algum ambiente e que é capaz de ações

autônomas neste ambiente como forma de alcançar os seus objetivos definidos em sua modelagem.

Por definição, um conjunto de propriedades pode ser atribuído aos agentes de software:

- **Autonomia:** capacidade de decidir suas ações sem intervenção externa, por exemplo, por um usuário.
- **Reatividade:** os agentes são capazes de perceber o ambiente em que estão inseridos e reagir em tempo hábil e de forma adequada para satisfazerem os objetivos que foram modelados.
- **Pró-Atividade:** os agentes são capazes de exibir comportamentos baseados em metas, tomando iniciativas para a realização dos seus objetivos.
- **Sociabilidade:** os agentes são capazes de interagir com outros agentes do sistema e através desse relacionamento buscar maneiras de atingir seus objetivos propostos.
- **Situados em um ambiente:** agentes estão inseridos e situados em um ambiente, que é definido em [Weyns, 2006] como segue: *“O ambiente é uma abstração de primeira classe que provê as condições externas necessárias para os agentes existirem e que media tanto as interações entre os agentes como o acesso aos recursos.”*

Por abstração de primeira de classe entende-se que o ambiente é um bloco de construção que encapsula suas próprias responsabilidades e ações, independentemente dos agentes. O ambiente provê as condições externas necessárias por servir como um meio de acesso para os agentes perceberem e atuarem sobre as entidades externas e os recursos, além de permitir a interação entre os agentes. Esta mediação além de permitir a atuação dos agentes, também serve como mecanismo de regulação com base em regras do domínio da aplicação e em outras limitações.

Um agente interpreta e armazena as situações percebidas em sua base do conhecimento. Este conhecimento do agente é a informação do que ele está ciente. Baseando-se no conhecimento atual e nos estados dos demais agentes, o agente decide suas ações. As ações são executadas no ambiente e são uma tentativa de modificar o ambiente em que cada agente está situado. Um agente não irá controlar todo o ambiente, ele terá, na melhor hipótese, controle sobre partes do ambiente. Logo, isto significa que cada agente tem uma posição no ambiente e tipicamente só tem controle sobre ela, assim como sua percepção fica limitada a uma região nos arredores da posição atual. Esta região limitada é conhecida como a vizinhança do agente.

Quando um agente atua no ambiente, nenhuma garantia de efeito é dada. A mesma ação executada duas vezes diante de circunstâncias aparentemente idênticas pode resultar em efeitos totalmente distintos, ou até mesmo o efeito esperado pode falhar [Wooldridge, 2002]. Desta forma, os agentes assumem sua presença num ambiente não determinístico o que inerentemente torna viável a adoção de agentes para domínios de sistemas dinâmicos e incertos.

O problema chave na adoção dos agentes reside na decisão de quais ações eles devem tomar para alcançarem da melhor forma os objetivos modelados, e contribuir eficientemente ao comportamento do sistema. O algoritmo interno de decisão e a estrutura do agente são tipicamente chamados de arquitetura. A arquitetura dos agentes é uma arquitetura de software para sistemas de decisão [Wooldridge, 2002], que resolve o problema de seleção da ação a ser tomada. Existem diversas arquiteturas possíveis, entre elas estão [Weyns, 2006],[Wooldridge, 2002]:

- Agentes baseados em raciocínio são uma abordagem da qual o agente possui uma representação simbólica do ambiente e do comportamento desejado, e através desta representação raciocina e elabora planos. Estes planos são uma série de ações a serem executadas. Normalmente, os agentes são provadores de teoremas em representações baseadas em lógica ou seguem a abordagem de

raciocínio humano, composta de crenças, desejos e intenções (BDI) [Braubach et al., 2004].

- Agentes reativos possuem enfoque na interação dinâmica com o ambiente. Internamente, os agentes são compostos de um sistema que transforma diretamente percepções em ações, permitindo reações em tempo real. Esta abordagem totalmente diferente da anterior foi proposta por conta da inviabilidade de certos agentes dotados de raciocínio serem impraticáveis para situações onde o ambiente requer uma alta dinamicidade.
- Agentes híbridos integram a arquitetura dos agentes com processos de raciocínio interno e planejamento com as características de reatividade, unindo as vantagens de planejamento a respostas em tempo hábil.

2.1.2

Sistemas Multi-Agentes

Sistemas multi-agentes são sistemas onde múltiplos agentes trabalham juntos a fim de realizarem uma funcionalidade do sistema. Esses agentes exploram o ambiente para a troca de informações e coordenação do seu comportamento. Sendo assim, o principal desafio entre sistemas multi-agentes e um simples agente reside na coordenação, que é definida em [Malone et al., 1994] como “*a gerência da dependência entre atividades*”. No âmbito dos agentes, as atividades são as ações que eles executam. Como já mencionado, os agentes cooperam entre si para alcançarem a meta do sistema e esta cooperação é determinada principalmente pelas dependências entre as ações dos agentes. Por exemplo, se dois agentes estão tentando se mover para a mesma localização, então as ações “move” de cada agente serão dependentes uma da outra, pois apenas uma poderá ter sucesso. Assim, os agentes devem se coordenar e negociar qual agente poderá se mover e qual deverá esperar.

A coordenação é primariamente realizada através da interação, ou seja, da comunicação entre os agentes. Essa comunicação pode ser direta ou indireta e a diferença entre esses tipos de comunicação é que na comunicação direta as

mensagens são enviadas diretamente ao receptor, enquanto que na comunicação indireta é realizada mediante o compartilhamento do ambiente. Assim, nesta última, um agente modifica o ambiente e os demais respondem a esta modificação e o modificam em resposta. Essa última etapa é tipicamente assíncrona e a manipulação do ambiente geralmente envolve o uso de sinais (marcas). Na biologia este processo é conhecido como *stigmergy* [Grass, 1959].

O uso de mecanismos de coordenação descentralizada é um dos objetivos deste trabalho. A descentralização implica em um estilo de coordenação em que os componentes da aplicação cooperam entre si, e nenhum componente tem o controle global do sistema [Schelfhout, 2006]. Por outro lado, no estilo centralizado de coordenação existe um componente que é conceitualmente o mestre, sendo os demais componentes escravos. Esta abordagem tipicamente resulta em um controle hierárquico. A coordenação descentralizada está diretamente relacionada com o fenômeno emergente e os sistemas auto-organizáveis, que serão descritos na próxima seção. Os sistemas multi-agentes é uma tecnologia propícia ao desenvolvimento de sistemas auto-organizáveis, pois dão o suporte necessário para o comportamento autônomo e a coordenação descentralizada.

2.2 Auto-Organização

Auto-organização é uma forma atraente para lidar com os requisitos dinâmicos do sistema. Refere-se a um processo em que o sistema modifica sua organização interna para se adaptar às mudanças em seus objetivos e no ambiente, sem um controle externo explícito. Da auto-organização geralmente surge um comportamento emergente que pode ser tanto desejável como indesejável. Devido ao dinamismo e abertura (*openness*) dos atuais sistemas multi-agentes, e do crescimento da distribuição, complexidade e das mudanças dinâmicas nos requisitos, o interesse por modelos e soluções auto-organizáveis vem crescendo.

Também é necessário distinguir entre sistemas em que não há controle interno e externo explícito daqueles em que há um controle interno centralizado. Por exemplo, nas sociedades de cupins, os diferentes arcos, que são a principal

estrutura de seus ninhos, estão todos localizados a uma mesma distância da rainha, por conta de um gradiente de feromônios. A rainha difunde essa informação, portanto trata-se de um controle interno. As seguintes definições foram feitas por Serugendo [Serugendo et al., 2005]:

Sistemas auto-organizáveis fortes são aqueles onde não há qualquer tipo de controle explícito, seja interno ou externo.

Sistemas auto-organizáveis fracos são aqueles onde, de um ponto de vista interno, há uma re-organização mediante a ação de controle interno central ou planejado.

2.2.1

Propriedades do comportamento da auto-organização

O comportamento dos sistemas auto-organizáveis é caracterizado pelas propriedades seguintes (mandatórias ou opcionais):

- **Ausência de um controle externo explícito.** Esta é uma propriedade mandatória que indica que o sistema é autônomo; que impõe mudanças e que sua organização é baseada exclusivamente nas decisões internas e não segue qualquer comando externo para realizar uma (re-) organização. Esta propriedade se refere à parte auto- de auto-organização.
- **Controle descentralizado.** Um sistema auto-organizável pode trabalhar sob controle descentralizado. Neste caso, não há nenhuma autoridade central interna ou fluxo de informação centralizado. Como resultado, o acesso à informação global é limitado pelas interações locais, que são governadas por regras simples. Essa propriedade geralmente não é mandatória, até porque podemos observá-la em sistemas auto-organizáveis naturais, como o já mencionado caso dos cupins. Entretanto, neste trabalho, os esforços estarão voltados para os sistemas multi-agentes auto-organizáveis fortes, em que a existência de um controle descentralizado é considerada obrigatória.

- **Operação dinâmica.** Esta propriedade mandatória está associada à evolução do sistema no tempo. Partindo do fato de que a organização evolui independentemente de qualquer controle externo, essa propriedade implica na continuidade do processo de auto-organização.

2.2.2

Fenômeno Emergente

Tipicamente, emergência é descrita como o fenômeno em que o comportamento global surge da interação entre partes locais do sistema. Exemplo diário deste fenômeno são os caminhos construídos de feromônios que surgem das interações locais das formigas com o ambiente, seguindo um caminho e depositando feromônios, conforme ilustra a Figura 1 [De Wolf et al., 2006].

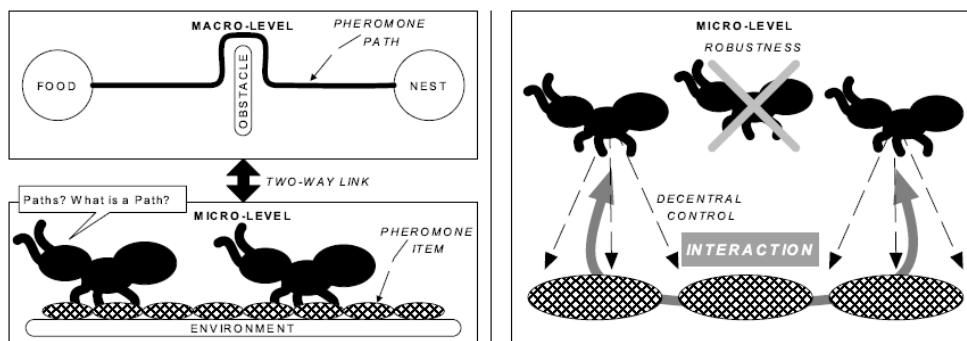


Figura 1 - Caminhos de Feromônio das Formigas

Não existe uma definição comum para o fenômeno emergente. Contudo, neste trabalho, adotaremos a definição dada por [De Wolf et al., 2006], que foi construída através de seu levantamento dos trabalhos existentes. A definição proposta é a seguinte:

Um sistema exhibe emergência quando há partes emergentes no nível macroscópico que surge das interações entre as partes no nível microscópico. Essas partes emergentes são novas no que diz respeito às partes do sistema.

A definição acima usa o conceito de emergência como um termo geral para denotar o resultado que surge: propriedades, comportamento, estruturas, padrões, etc. O nível mencionado refere-se a certos pontos de vista. No nível

macroscópico, o sistema é considerado como um todo, já no nível microscópico o sistema é observado do ponto de vista das entidades individuais que compõem o sistema. Esta definição, segundo o autor, é resultado de um estudo que inventariou as principais características encontradas na literatura.

Certa propriedade, comportamento, estrutura, etc. é dita emergente quando é necessária outra ontologia para descrevê-la, diferente da usada para descrição da estrutura interna do sistema. Por exemplo, um caminho de feromônios não faz parte da ontologia da estrutura do sistema que consiste de formigas, pedaços de feromônio, ninho e comida, exemplo exibido na Figura 1. Ainda, emergência é sempre considerada em concordância ao que foi definido como nível microscópico e macroscópico. Por exemplo, podemos considerar todo o sistema desempenhado pelas formigas como um módulo em um sistema de larga escala, então o sistema das formigas seria considerado um módulo do nível microscópico. As propriedades macroscópicas internas deste módulo são microscópicas em relação a todo o sistema, e as propriedades macroscópicas são então consideradas como resultado das interações entre os demais módulos.

2.3

Meta-modelo Agents and Artifacts (A&A)

Como já mencionamos, sistemas multi-agentes são modelados como uma composição de agentes, vistos como entidades autônomas situadas em um ambiente, que podem tanto atuar no nível físico como computacional, e que interagem uns com os outros e com os recursos do ambiente para realizarem metas individuais ou do grupo. Tradicionalmente, o ambiente consiste no contexto de implantação que fornece serviços de comunicação e acesso a recursos físicos. Desta forma, os engenheiros de sistemas multi-agentes constroem os agentes enquanto o ambiente é apenas uma saída da etapa de análise. Entretanto, em trabalhos recentes, o ambiente tem sido reconhecido como um aspecto relevante na modelagem dos sistemas multi-agentes, pois eles podem abstrair parte da complexidade do sistema através de seus serviços, liberando os agentes de responsabilidades paralelas a suas.

No meta-modelo A&A, os sistemas multi-agentes são modelados a partir de duas abstrações: agentes e artefatos [Omicini et al., 2005][Ricci et al., 2006]. Os agentes são entidades autônomas e proativas que encapsulam seus controles e suas metas. Contudo, algumas entidades não necessitam de autonomia nem de proatividade para serem caracterizadas. Tipicamente, essas entidades servem como ferramentas que oferecem funcionalidades específicas. No modelo A&A, essas últimas entidades são chamadas de artefatos. Os artefatos são elementos passivos e reativos que fornecem serviços aos agentes através de uma interface. Desta forma, os artefatos podem ser usados como forma de mediar as interações com os agentes, suportando mecanismos de coordenação e incorporar atividades realizadas pelo ambiente.

Neste trabalho, nos baseamos no meta-modelo A&A para a construção do ambiente provido pelo framework. Desta forma, o ambiente é composto por agentes baseados no conceito de artefatos, que fornecem serviços de armazenamento de informações em posições específicas do ambiente e permitem a realização dos padrões que atuam diretamente na área ocupada, utilizando o ambiente como mecanismo de coordenação. Mais detalhes são expostos no Capítulo 4.

2.4

Padrões básicos de auto-organização

Nesta seção, descrevemos quatro dos cinco padrões básicos documentados em [Gardelli et al., 2007], o quinto padrão não descrito foge um pouco do escopo dos mecanismos de coordenação, tratando-se de um padrão de distribuição ordenada. Gardelli descreveu os padrões de acordo com [Lind, 2003], contudo, adicionou novas propriedades na linguagem de descrição que englobam aspectos essenciais da auto-organização. O esquema é mostrado na tabela a seguir. No framework, esses quatro padrões são disponibilizados em uma estrutura, permitindo seu pronto uso, ou seja, diminuindo o nível de abstração. Através da implementação por meio de capabilities do Jadex (próxima seção), o framework disponibiliza os padrões como mecanismos de coordenação a serem importados

(em nível de código) pelos agentes da aplicação. Os padrões são: Replication, Evaporation, Aggregation e Diffusion.

Tabela 1 - Esquema de Descrição dos Padrões

Nome	O nome do padrão
Pseudônimo	Nomes alternativos
Problema	O problema resolvido pelo padrão
Forças	Trade-offs na dinâmica do sistema
Entidades	Entidades participantes do padrão
Dinâmica	Interação entre as entidades
Feedback Loop	Interações responsáveis pelo feedback loop
Localidade	Descreve o tipo de localização requerida
Dependências	Requisitos do ambiente
Exemplos	Um exemplo abstrato de uso
Implementação	Formas de implementar
Casos de uso	Aplicações que usam o padrão
Consequências	Efeitos na modelagem do sistema

Os exemplos dos padrões serão mostrados através da representação apresentada na Figura 2. Nesta representação, as letras (A,B,C,D e E) ilustram os artefatos do ambiente, os traços que interligam os artefatos representam os segmentos de interconexão, as várias formas contidas nos artefatos representam diferentes tipos de informações.

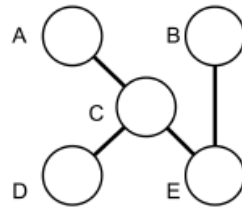


Figura 2 - Representação Abstrata de uma Topologia de um SMA

2.4.1 Replication

O padrão Replication é normalmente encontrado em sistemas naturais como forma de aumentar a sua segurança e robustez. Por exemplo, todas as células do corpo humano mantêm uma cópia local do DNA, permitindo a elas a possibilidade de se recuperar de pequenas mutações. Este mecanismo também é comumente aplicado em discos rígidos de servidores como forma de evitar perdas, técnica conhecida como Redundant Array of Independent Drives (RAID). É relevante mencionar que na propriedade Consequências, Gardelli afirma que o padrão Replication não funciona em conjunto com o padrão Diffusion. Porém, Gardelli parte do princípio que todas as localidades do ambiente executam os mesmo padrões. Contudo, no JASOF, como será visto no Capítulo 4, cada agente do ambiente pode desempenhar diferentes padrões em sua localidade, assim, permitindo a ação em conjunto dos padrões Replication e Diffusion.

Tabela 2 - Resumo das Propriedades do Padrão Replication

Nome	Replicação/Replication
Pseudônimo	Duplicação, Redundância
Problema	1 – Como diminuir o tempo de acesso à informação? 2 – Em caso de ataque ou falha, como evitar a perda de informações?
Forças	Replicação garante melhor segurança e rápido acesso pelo preço de maior consumo de memória.

Entidades	Participam do padrão: artefatos, agentes do ambiente e agentes usuários.
Dinâmica	O agente usuário armazena informação em um artefato, que mantém dados de time-stamp. Agentes do ambiente monitoram os artefatos por novas informações, que são eventualmente enviadas pelos vizinhos.
Feedback Loop	-
Localidade	Atua por todo o ambiente.
Dependências	Requer um ambiente em conformidade com o meta-modelo A&A.
Exemplos	Figura 3
Implementação	Os agentes do ambiente devem realizar periodicamente uma inspeção ou serem acionados por um gatilho
Casos de uso	1 – Memória Cache. 2 – RAID. 3 – Infraestrutura de Grids.
Consequências	Replication pode não funcionar quando combinado com outros padrões que espalham informações pelo ambiente.

Conforme ilustra a Figura 3, do estado inicial (a), o agente do ambiente irá replicar a informação contida exclusivamente no artefato E para todo o ambiente. Assim, após a aplicação do padrão Replication, todos os artefatos conterão a informação replicada, conforme ilustra o estado final (b).

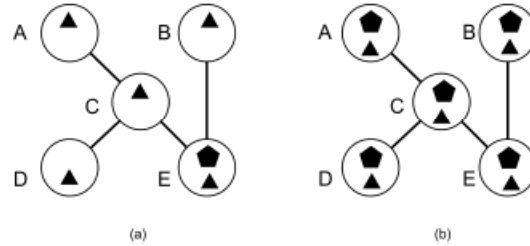


Figura 3 - Padrão Replication: (a) estado inicial; (b) estado final desejado

2.4.2 Evaporation

Em colônia de insetos, a coordenação é geralmente obtida pelo uso de substâncias químicas, usualmente na forma de feromônio. Esses feromônios atuam como marcadores para a realização de tarefas específicas, e são regulados por processos do ambiente, como agregação, difusão e evaporação. Cada um desses processos tem sua própria relevância, por isso são analisados separadamente como padrões. Esse tipo de mecanismo de coordenação indireta, mediado pelo ambiente, é chamado de *Stigmergy* [Grass, 1959].

A evaporação é um processo que pode ser observado diariamente, embora com diferentes implicações. Por exemplo, a partir da intensidade de um perfume sentindo é possível deduzir a quantidade e a distância da fonte. No caso das formigas, a alta concentração de feromônio pode indicar um caminho a uma fonte de comida recente, por outro lado, sua presença reduzida pode significar que a fonte está esgotada ou que nada foi encontrado. O intuito do padrão Evaporation é minimizar a possibilidade de sobrecarga de informações no ambiente, baseando-se num critério de tempo. A Figura 4 exhibe a dinâmica da evaporação. A partir do estado inicial (a), decorrido um período de tempo, os artefatos evaporam a informação armazenada, chegando ao estado final (b).

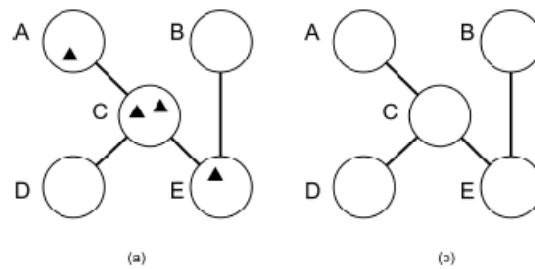


Figura 4 - Padrão Evaporation: (a) estado inicial; (b) estado final desejado

Tabela 3 - Resumo das Propriedades do Padrão Evaporation

Nome	Evaporação/Evaporation
Pseudônimo	-
Problema	O ambiente pode rapidamente ficar sobrecarregado pelas informações deixadas pelos agentes.
Forças	Taxa de evaporação muito alta libera mais memória, mas requer mais tempo de processamento. Ainda, informação removida não pode ser recuperada.
Entidades	Participam do padrão: artefatos, agentes do ambiente e agentes usuários.
Dinâmica	O agente usuário armazena informação em um artefato, que mantém dados de timestamp e um contador. Agentes do ambiente apagam as informações obsoletas que alcançaram o valor zero no contador.
Feedback Loop	Agentes usuários depositam informações no ambiente enquanto os agentes do ambiente as evaporam.
Localidade	Percepções e ações ocorrem apenas localmente.
Dependências	Requer um ambiente em conformidade com o meta-modelo A&A.
Exemplos	Figura 4
Implementação	Os agentes do ambiente devem realizar periodicamente uma inspeção ou serem acionados por um gatilho

Casos de uso	Um elemento fundamental no processo <i>Stigmergy</i> e nas aplicações baseadas em feromônios digitais. [Mamei et al., 2005]
Consequências	-

2.4.3 Aggregation

O padrão Aggregation é um mecanismo de reforço, também observado em atividades humanas. Por exemplo, em um sistema de recomendações, quanto mais qualificações positivas se recebe sobre um determinado serviço ou produto, maior será a crença e relevância do serviço ou produto quando necessitado. O padrão Aggregation atua espontaneamente, ou seja, ao ser adicionada uma informação no ambiente esta sofre uma agregação de imediato, onde informações semelhantes são unidas e percebidas como uma única informação, porém, com um fator de relevância maior. Quando usado com o padrão Evaporation, o padrão Aggregation permite ao desenvolvedor criar um mecanismo semelhante a um feedback loop positivo/negativo.

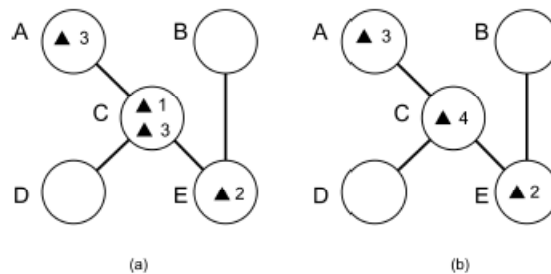


Figura 5 - Padrão Aggregation: (a) estado inicial; (b) estado final desejado

Tabela 4 - Resumo das Propriedades do Padrão Aggregation

Nome	Agregação/Aggregation
Pseudônimo	-
Problema	SMAs de grande porte sofrem pela grande quantidade de informação depositada pelos agentes, que deve ser tratada para produzir

	informações relevantes.
Forças	Alto grau de agregação resulta em informação que reproduzem o status atual do ambiente, entretanto requer mais processamento.
Entidades	Participam do padrão: artefatos, agentes do ambiente e agentes usuários.
Dinâmica	O agente usuário armazena informação em um artefato. Agente usuário procura por novas informações e agrega com as informações já existentes para produzir um resultado coerente.
Feedback Loop	Agentes usuários depositam informações no ambiente enquanto que os agentes do ambiente a sintetizam em uma informação agregada com as demais.
Localidade	As percepções dos agentes e suas ações só ocorrem localmente, enquanto os agentes do ambiente atuam em pelo menos um salto de distância.
Dependências	Requer um ambiente em conformidade com o meta-modelo A&A.
Exemplos	Figura 5
Implementação	Os agentes do ambiente devem realizar periodicamente uma inspeção ou serem acionados por um gatilho

Casos de uso	Um elemento fundamental no processo <i>Stigmergy</i> e nas aplicações baseadas em feromônios digitais. [Mamei et al., 2005]
Consequências	-

2.4.4 Diffusion

Em sistemas naturais, quando o feromônio é depositado no ambiente, espontaneamente o mesmo tende a se difundir pelas localidades vizinhas; este processo é conhecido como difusão [Bonabeau et al., 1999]. Com exemplo, tem-se o perfume exalado por esses feromônios, que possui um determinado alcance e é difundindo igualmente nas localidades ao redor. Partindo de um estado qualquer, a difusão eventualmente distribui a informação igualmente por todos os nós, provendo um sistema de média distribuída.

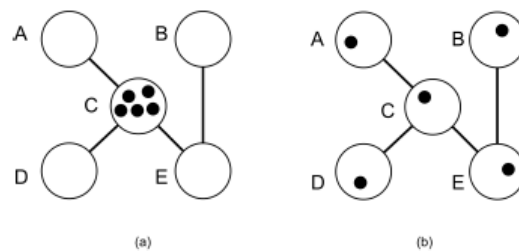


Figura 6 - Padrão Diffusion: (a) estado inicial; (b) estado final desejado

Conforme ilustra a Figura 6, a partir do estado inicial (a), o artefato C difunde a informação igualmente por todo o ambiente, finalizando no estado final (b), onde cada artefato possui uma porção do item difundido. Enquanto os padrões Aggregation e Evaporation atuam localmente, o padrão Diffusion e Replication requerem uma noção de topologia.

Tabela 5 - Resumo das Propriedades do Padrão Diffusion

Nome	Difusão/Diffusion
Pseudônimo	Plain Diffusion, Osmosis.

Problema	Em SMA onde os agentes têm percepções limitadas do ambiente, o processo de raciocínio pode ficar inviável pela falta de informações.
Forças	Uma difusão com raio de amplitude maior pode levar informações para agentes distantes da fonte, porém, como consequência, a carga da infraestrutura fica comprometida, tanto em tempo de máquina como em memória.
Entidades	Participam do padrão: artefatos, agentes do ambiente e agentes usuários.
Dinâmica	O agente usuário armazena informação em um artefato. A informação recebe um peso. Os agentes do ambiente difundem a informação decrementando este peso.
Feedback Loop	Os agentes usuários depositam a informação no ambiente enquanto os agentes do ambiente a espalham pelas localizações vizinhas.
Localidade	As percepções dos agentes e suas ações só ocorrem localmente, enquanto os agentes do ambiente atuam em pelo menos um salto de distância.
Dependências	Requer um ambiente em conformidade com o meta-modelo A&A.

Exemplos	Figura 6
Implementação	Os agentes do ambiente devem realizar periodicamente uma inspeção ou serem acionados por um gatilho
Casos de uso	Um elemento fundamental no processo <i>Stigmergy</i> e nas aplicações baseadas em feromônios digitais. [Mamei et al., 2005]
Consequências	Diffusion pode não funcionar em conjunto com outros padrões que espalham informações pelo ambiente.

2.5 Jadex

Jadex é um framework para construção de agentes que proporciona a adição de capacidades cognitivas, de maneira fácil e através do modelo BDI, combinado com técnicas de engenharia de software como XML e Java.

Na Figura 7 uma visão geral da arquitetura abstrata do Jadex é apresentada. Do ponto de vista externo, o agente é uma caixa preta, que recebe e envia mensagens. Mensagens recebidas, eventos internos e novas metas servem como entrada (*input*) para a o mecanismo de reação e deliberação (*Reaction Deliberation*). Então, baseado nesses resultados da deliberação, o mecanismo direciona os eventos para os planos que já estão em execução ou aos que possam ser acionados a partir da biblioteca de planos. Os planos, quando em execução, podem modificar a base de conhecimento dos agentes, enviar mensagens para outros agentes, criar novas sub-metas e ainda ativar eventos internos. O mecanismo de reação e deliberação é geralmente igual para todos os agentes. Entretanto, o comportamento de um agente em específico é determinado pelas

suas crenças (*Beliefs*), metas (*Desires*) e planos (*Intentions*). A seguir, cada um desses conceitos serão descritos em detalhes.

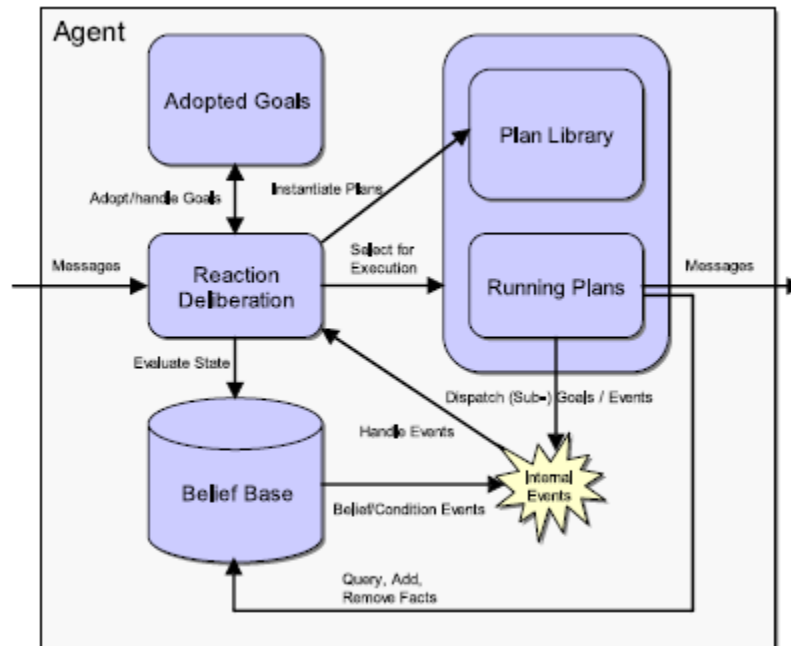


Figura 7 - Arquitetura Abstrata do Jadex

- **Crenças:** Para o Jadex as crenças correspondem à base do conhecimento do agente. Nessa base, objetos comuns da linguagem Java podem ser armazenados como fatos, que são crenças conhecidas, ou como um conjunto de fatos. Através dos nomes das crenças, a base do conhecimento pode manipular esses objetos, seja definindo novos fatos, adicionando ou removendo. As crenças podem ser usadas como entrada para a máquina de raciocínio, como pré-condições para a criação de planos ou metas. Essas crenças são sempre monitoradas, e na ocorrência de uma alteração relevante os planos e metas são ajustados.
- **Metas:** As metas são os desejos que os agentes buscam realizar. Para alcançá-los os agentes executam as ações definidas nos planos. As metas no Jadex podem ser de vários tipos. O tipo *perform goal* está relacionado à execução imediata de certas ações. Portanto, o objetivo

é considerado atingido quando as ações forem executadas, independentemente do resultado dessas ações. O *achieve goal* atua no sentido tradicional, ou seja, o objetivo é atingir a um resultado desejado, sem especificar como alcançá-lo. Assim, os agentes podem tentar vários planos diferentes em busca do resultado determinado. A *query goal* é semelhante à meta anterior, o objetivo é atingir um objetivo, porém, seu resultado não é definido com um estado, mas como algumas informações que o agente quer conhecer. Por fim, para o tipo *maintain*, o agente procura manter o estado desejado, e continuamente executa planos adequados para restabelecer o estado sempre que necessário.

- **Planos:** Os planos representam os elementos comportamentais de um agente e são compostos por duas partes: *head* e *body*. A especificação da primeira parte está relacionada às circunstâncias que levam um plano a ser selecionado, como, por exemplo, o recebimento de um tipo de mensagem e o alcance de um objetivo. Na segunda parte, é a elaboração do conjunto de ações a serem executadas em busca do objetivo a ser alcançado.
- **Capacidade (Capability):** As capacidades representam o agrupamento dos elementos que compõem um agente BDI, como, por exemplo, suas crenças, metas, planos e eventos. As *capabilities* são utilizadas como meio de reusabilidade, pois podem ser importadas por outros agentes em seus arquivo de definição, e assim agregar os componentes definidos na *capability* ao agente.

```

01 <!--
02  A simple translation agent for translating words from English to German.
03 -->
04 <agent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
05  xsi:noNamespaceSchemaLocation="http://jadex.sourceforge.net/jadex.xsd"
06  name="ta"
07  package="jadex.examples.tutorial">
08
09  <imports>
10  <import>jadex.util.*</import>
11  </imports>
12
13  <plans>
14  <plan name="egtrans">
15  <constructor>new EnglishGermanTranslationPlan()</constructor>
16  <filter>EnglishGermanTranslationPlan.getEventFilter()</filter>
17  </plan>
18  </plans>
19
20  <beliefs>
21  <beliefset name="egwords" class="Tuple">
22  <fact>new Tuple("milk", "Milch")</fact>
23  <fact>new Tuple("cow", "Kuh")</fact>
24  <fact>new Tuple("cat", "Katze")</fact>
25  <fact>new Tuple("dog", "Hund")</fact>
26  </beliefset>
27  </beliefs>
28
29  <expressions>
30  <expression name="query_egword">
31  SELECT ANY Swordpair.get(1)
32  FROM Swordpair in $beliefbase.egwords
33  WHERE Swordpair.get(0)=Seword
34  </expression>
35  </expressions>
36
37 </agent>

```

Figura 8 - Exemplo de um Arquivo de Definição de um Agente (ADF)

Para criar e iniciar um agente, o sistema precisa conhecer as propriedades do agente para que ele possa ser instanciado. O estado inicial do agente é determinado, entre outros, por suas crenças, objetivos e da biblioteca de planos conhecidos. Jadex usa uma abordagem declarativa e procedural para definir os componentes de um agente. Os planos devem ser implementados em Java e precisam estender uma classe específica do Jadex. Todos os outros conceitos são especificados usando a linguagem XML, seguindo o meta-modelo especificado em XML Schema, permitindo a criação de agentes de forma declarativa. Esta descrição é chamada *Agent Definition File* (ADF), um exemplo deste arquivo pode ser visto na Figura 8, que exhibe um agente com fatos em sua base e um plano de tradução.