

## 2 Fundamentação teórica

Este capítulo apresenta uma breve introdução teórica aos principais conceitos utilizados na elaboração do método de verificação experimental de sistemas multiagentes auto-organizáveis que será apresentado no próximo capítulo deste trabalho. Para uma noção mais detalhada sobre os temas e assuntos aqui introduzidos, recomenda-se a leitura de bibliografias complementares.

Em linhas gerais, a seção 2.1 deste capítulo descreve as principais características da computação autônoma, paradigma de desenvolvimento de *software* proposto pela IBM (IBM, 2001). Em seguida, a seção 2.2 apresenta uma breve introdução aos sistemas multiagentes, tecnologia mais difundida para a engenharia de sistemas auto-organizáveis. Na seção 2.3 discutem-se as dificuldades introduzidas pela presença do comportamento emergente nos sistemas auto-organizáveis.

### 2.1 Computação autônoma

O constante aumento da complexidade dos sistemas computacionais trouxe consigo uma característica previsível: quanto mais complexo for um sistema, maior será a dificuldade de operá-lo manualmente. Para solucionar esse problema, duas alternativas podem ser analisadas: a primeira é tentar reduzir a complexidade do sistema, tornando mais simples a sua operação. Logicamente, isso nem sempre é aplicável, pois na maioria das vezes a complexidade de um sistema é intrínseca ao processo que ele implementa. A segunda alternativa é a concepção de técnicas de execução autônoma desse sistema, para que ele dependa cada vez menos da intervenção dos seus administradores e usuários.

Entretanto, a execução autônoma de um sistema implica que ele seja capaz de realizar tarefas sem a intervenção externa, seja de outros sistemas ou mesmo dos administradores e usuários. No âmbito industrial, por exemplo, um sistema que realiza a gestão do estoque de uma fábrica poderia ser capaz de prever, com certa antecedência, a falta de uma determinada matéria-prima e, então, sugerir ao gerente que ela fosse encomendada já no próximo pedido.

Num cenário ainda mais avançado, esse mesmo sistema poderia realizar automaticamente a cotação do pedido junto aos diversos fornecedores cadastrados e apresentar a alternativa mais viável ao gerente, levando em consideração fatores como: preço, prazo de entrega, confiabilidade do fornecedor, etc.

Foi com o propósito de estimular discussões sobre possíveis soluções para a execução autônoma dos sistemas computacionais que, em meados de 2001, a IBM apresentou um manifesto indicando que o principal obstáculo para o progresso da Tecnologia da Informação era realmente o aumento desenfreado da complexidade dos sistemas de *software* e, portanto, era necessário adotar estratégias e arquiteturas autonômicas. Esse novo paradigma de projeto e desenvolvimento de *software* foi, então, chamado de **computação autonômica** (IBM, 2001).

Tipicamente, o objetivo principal de um sistema autonômico ser capaz de se auto-gerenciar a partir de alguns objetivos de alto-nível definidos por seus administradores (Kephart, 2003), atendendo à quatro propriedades básicas:

- **Autoconfiguração:** Sugere a capacidade do sistema constantemente configurar si próprio e seus componentes, mesmo em vista de possíveis alterações;
- **Autodiagnóstico e cura:** Define a competência do sistema para detectar e corrigir possíveis falhas em sua execução;
- **Auto-otimização:** Indica o ato de o sistema monitorar e controlar continuamente os seus próprios recursos a fim de garantir um funcionamento ideal com relação aos requisitos definidos;
- **Autoproteção:** Enfatiza a capacidade de um sistema detectar e se proteger contra ataques maliciosos ou falhas em cascata, mantendo intactas a sua segurança e integridade.

Do ponto de vista arquitetural, esses sistemas podem ser modelados como coleções de elementos autonômicos – indivíduos constituintes do sistema, que contém recursos e que entregam serviços aos usuários ou à outros elementos autonômicos (Kephart, 2003). O autogerenciamento global do sistema emergirá mais das interações entre esses elementos autonômicos do que do controle interno de cada um deles – assim como a inteligência de uma colônia de formigas é oriunda da inter-relação entre as formigas que a compõe.

A figura 2.1 descreve a estrutura interna de um sistema autonômico. Pode-se perceber que cada um dos seus elementos é composto por um gerenciador autonômico e um ou mais elementos gerenciados (um recurso de hardware – memória, processador, impressora, etc. – ou de *software* – banco

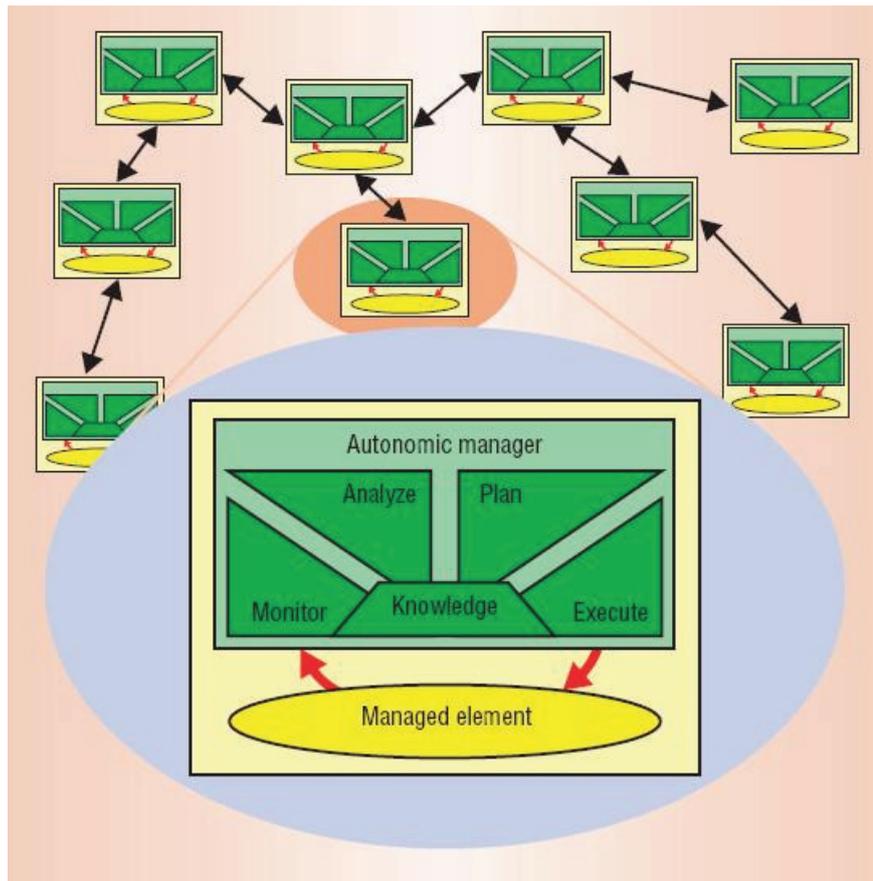


Figura 2.1: Estrutura interna de um elemento autônomo (Kephart, 2003)

de dados, sistema externo, etc.). Em destaque, é possível ainda identificar os principais módulos de um gerenciador autônomo: o módulo de **monitoração** capta mensagens e eventos gerados no ambiente; o módulo de **análise** avalia os eventos percebidos; o módulo de **planejamento** estabelece um plano de ações em resposta aos eventos analisados; e, finalmente, o módulo de **execução** executa as ações previstas neste plano. Todos os módulos internos ao gerenciador autônomo consultam e alteram a sua base de conhecimento, mantendo-a atualizada para cada situação.

Através da monitoração dos elementos gerenciados e a elaboração e posterior execução dos planos de ações obtidos através da análise das informações coletadas, o gerenciador autônomo alivia a responsabilidade dos usuários ou administradores do sistema de diretamente monitorar esses recursos.

## 2.2

### Sistemas multiagentes

Sistemas multiagentes são sistemas compostos por múltiplos elementos computacionais interativos, conhecidos como agentes. Os agentes, por sua vez, são sistemas computacionais que possuem duas capacidades importantes:

Primeiro, eles são capazes de agir de forma autônoma, isto é, de decidir o que é necessário para satisfazer os seus objetivos de projeto. Segundo, eles são capazes de interagir com outros agentes, não apenas trocando informações, mas envolvendo-se em atividades sociais de cooperação, coordenação, negociação, etc. (Wooldridge, 2002).

Apesar de ser um campo de pesquisa relativamente novo, o número de trabalhos envolvendo sistemas multiagentes vem aumentando consideravelmente com o passar dos anos. Segundo (Wooldridge, 2002), isso pode ser parcialmente explicado pelo fato de que agentes são considerados um paradigma de *software* adequado para explorar os problemas que envolvem sistemas distribuídos de larga escala, como a Internet, onde a interação entre os seus componentes é de grande importância para o sistema como um todo.

### 2.2.1 Agentes

Um agente é basicamente um sistema computacional que reside em um ambiente e que é capaz de tomar decisões e executar ações sem interferência externa, com a finalidade de atingir os objetivos para os quais foi projetado. É possível, ainda, que, para tomar as decisões mais adequadas, esse agente precise interagir com outros agentes ou com o próprio ambiente em processos de cooperação, coordenação, negociação, etc. Essa interação se dá por meio dos sensores – que captam e percebem eventos do ambiente – e dos efetadores – que atuam sobre o ambiente a fim de modificá-lo.

Sendo assim, algumas propriedades intrínsecas a todo e qualquer agente podem ser enumeradas (Jennings, 2001):

- **Autonomia:** Um agente é, por definição, um sistema autônomo. Isso significa que ele é capaz de agir sem a interferência do usuário ou de outro sistema externo;
- **Reatividade:** Agentes são capazes de perceber eventos disparados no ambiente e responder a estes eventos de modo a atingir seus objetivos de projeto (comportamento reativo);
- **Pró-atividade:** Agentes também são capazes de exibir um comportamento orientado a objetivo, tomando a iniciativa e agindo de forma a atingi-lo (comportamento pró-ativo);
- **Habilidade social:** Um agente é capaz de interagir com o ambiente ou com outros agentes em processos sociais muitas vezes complexos;

Outras propriedades também importantes são muitas vezes omitidas por alguns autores por não serem essenciais à existência de um agente, mas ainda assim são muito importantes e devem ser mencionadas:

- **Aprendizagem:** Um agente é capaz de aprender observando como o ambiente ou os outros agentes se comportam a cada nova situação enfrentada. Em muitos casos, tais observações levam à formação de uma base de conhecimento extremamente valiosa, que pode ajudar o agente a cumprir os objetivos que lhe foram atribuídos;
- **Mobilidade:** Agentes são muitas vezes capazes de se movimentarem dentro do ambiente e também entre diferentes ambientes, de acordo com suas necessidades.

Um agente não possui controle total sobre o ambiente no qual ele reside. Na melhor das hipóteses ele tem um controle parcial sobre tal (Wooldridge, 2002). Na prática, isso significa que um agente normalmente conhece e atua apenas sobre uma parte limitada do ambiente. Nesse contexto, os agentes móveis têm a vantagem de poderem se locomover, alterando a sua localização e buscando um melhor posicionamento a fim de contribuir para o alcance dos seus objetivos.

As ações que são executadas por um agente durante o seu ciclo de vida normalmente têm a finalidade de modificar o ambiente de acordo com os seus objetivos. Entretanto, um agente nunca tem a certeza do real efeito que uma ação terá sobre o ambiente quando executada. Uma mesma ação poderá, inclusive, ter efeitos diferentes quando realizada em momentos e situações distintas.

Do ponto de vista da arquitetura de sistema, agentes de *software* são normalmente projetados como máquinas de estados, onde cada estado contém o conhecimento do agente em um dado momento. A alternância de estados se dá por meio da execução de ações pelo agente, seja ela de forma reativa ou pró-ativa. Esse, inclusive, é o principal desafio enfrentado ao se projetar um agente: determinar quais ações devem ser executadas em cada situação para se atingir objetivos pré-determinados é, muitas vezes, uma tarefa complicada.

Nesse contexto, os agentes são normalmente divididos em categorias conforme suas características de construção e métodos para tomada de decisão (Wooldridge, 2002):

- **Agentes de raciocínio dedutivo:** Agentes que se enquadram nessa categoria buscam atingir seus objetivos através da elaboração e execução de um plano de ações. Este plano é composto por uma série de ações

que, de acordo com o conhecimento do agente, levarão o ambiente ao seu estado desejado. Tal sequência de ações é construída a partir de deduções lógicas realizadas sobre as representações simbólicas do ambiente e do seu estado desejado (objetivo). Basicamente, o raciocínio destes agentes se restringe ao ato de provar formalmente (dedução lógica) um teorema;

- **Agentes de raciocínio prático:** É sabido que nem todos os problemas são solucionáveis por meio de dedução lógica. Muitas vezes a dificuldade é formular uma representação lógica do problema a ser resolvido. Em outras, é necessário algo além da lógica para se chegar a uma solução. Dessa forma, agentes de raciocínio prático são similares aos agentes de raciocínio dedutivo, mas também levam em consideração seus desejos, valores e crenças no processo de tomada de decisão;
- **Agentes reativos:** As abordagens baseadas em raciocínio descritas acima se mostraram muitas vezes inviáveis de serem aplicadas quando os agentes possuem muitos objetivos, muitas vezes conflitantes, e operam em ambientes complexos e dinâmicos. [De Wolf, 2007]. Nesse caso, os agentes reativos são mais adequados, pois reagem em tempo real aos eventos percebidos no ambiente;
- **Agentes híbridos:** Esses agentes são, na verdade, uma junção entre agentes de raciocínio e agentes reativos. Eles integram a sua capacidade de raciocínio (lógico e/ou prático) com a característica de reatividade a estímulos do ambiente, combinando os benefícios de planejamento de ações e resposta em tempo-real (De Wolf, 2007);
- **Agentes baseados em comportamento:** Ao contrário dos agentes reativos, que possuem seu comportamento fixado em sua arquitetura, os agentes baseados em comportamento possuem papéis (funções). Esses papéis são conjuntos de ações relacionadas direcionadas a um objetivo específico. Um mesmo agente pode desempenhar diferentes papéis em um determinado momento, o que permite a modificação dos seus objetivos (e, conseqüentemente, do seu comportamento) em tempo real. Isso os torna mais adequados para ambientes complexos, onde o processo de tomada de decisão deve ser mais dinâmico e flexível.

### 2.2.2

#### Interação entre agentes

Apesar de um agente de *software* ser um sistema computacional inteligente, na maioria das vezes ele é incapaz de solucionar um problema mais

complexo sozinho. Geralmente, um sistema computacional eficiente é composto por subsistemas que interagem entre si com a finalidade de atingir um determinado objetivo. Há, ainda, a interação com sistemas ou entidades externas, também com esse mesmo objetivo. Os sistemas multiagentes não fogem a essa regra: o sucesso de uma solução multiagente depende da interação entre os agentes que o compõem.

Entretanto, para que se consiga a funcionalidade pretendida sem que o sistema atinja um estado caótico, é preciso haver organização no processo de interação entre os agentes, ou seja, coordenação na execução de ações que tenham o potencial para afetar outros agentes. Para isso é preciso que os agentes se comuniquem eficientemente, de forma direta (através do envio de mensagens) ou indireta (por meio da modificação do ambiente).

Esse processo de coordenação pode ser centralizado ou descentralizado. No primeiro, um único agente exerce o papel de coordenador e controla toda a execução do sistema. Não existe troca direta de mensagens entre os agentes, ou seja, o coordenador é, também, um intermediário em todas as comunicações. Já no segundo, não existe um coordenador; nesse caso, todos possuem responsabilidades de coordenação e comunicam-se diretamente com os outros agentes, o que é um fator complicador para se manter a organização do sistema. Sistemas multiagentes centralizados não fazem parte do escopo deste trabalho.

## 2.3

### Comportamento emergente

O processo de auto-organização presente na inter-relação entre os elementos que compõem um sistema auto-organizável muitas vezes resulta em comportamentos novos, não previstos, gerados pela interação entre os elementos autônomos do sistema e observáveis apenas no seu nível macroscópico. Esses comportamentos novos, não previstos, são denominados **comportamentos emergentes**.

É interessante observar que, apesar de cada agente possuir certo nível de raciocínio, a inteligência global do sistema não é alcançada individualmente. Ela costuma ser fruto da interação e comunicação dos agentes que o compõem. Na maioria das vezes, inclusive, ela é invisível quando se observa o sistema em seu nível microscópico. Pode-se afirmar, então, que a inteligência global de um sistema multiagente constitui um comportamento emergente clássico.

Em uma colônia de formigas, por exemplo, as unidades individuais (formigas) desempenham apenas tarefas triviais. Entretanto, considerando a colônia como um todo, percebe-se uma inteligência avançada proveniente da

inter-relação entre as formigas por meio dos caminhos de feromônio. E, nesse caso, essa inteligência que emerge da colônia é o principal fator responsável pelo seu sucesso, no que diz respeito à atividade de coleta de comida.

Todavia, é importante ressaltar que nem sempre esse comportamento emergente será benéfico. Quanto maior o sistema (em termos de complexidade, interconexões entre as suas unidades, distribuição geográfica, etc.), maior será a chance de ele apresentar, em algum momento, um comportamento emergente indesejado, prejudicial ao sistema como um todo. Dessa forma, um dos grandes desafios da engenharia de sistemas auto-organizáveis é fazer com que um comportamento dito emergente convirja para um padrão ou tendência desejável que, na maioria dos casos, é um objetivo ou sub-objetivo do sistema (Gatti, 2009).

Portanto, é importante que se tenha um mecanismo eficiente para garantir antecipadamente, isto é, antes mesmo que esse sistema seja colocado em produção, que ele realmente apresentará um padrão de comportamento esperado e desejado.