2
Fundamentação Teórica

Neste capítulo será feita uma discussão sobre os aspectos teóricos necessários para o embasamento do trabalho aqui apresentado. Os principais tópicos apresentados dizem respeito à gerência de redes e alguns dos mais importantes padrões de gerência criados por organizações de normatização. É através dos modelos e protocolos de gerência definidos por tais padrões que a solução proposta neste trabalho se torna efetivamente capaz de se comunicar com equipamentos de rede para identificar falhas e eventualmente corrigi-las. Também serão apresentados os principais paradigmas de gerência de redes concebidos ao longo do tempo. Em particular, a caracterização dos modelos centralizado e distribuído é bastante relevante para este trabalho. Em seguida será feita uma apresentação de aspectos teóricos relacionados a agentes de software. Explorar-se-á algumas de suas principais características, assim como a sua organização em sistemas multi-agentes.

2.1
Gerência de Redes

2.1.1
Introdução

Redes de telecomunicações são tipicamente compostas por um determinado número de nós geograficamente distribuídos, conectados entre si através de algum meio físico (como fibras ópticas, cabos coaxiais ou enlaces de rádio) que permitem o fluxo de informações entre dois ou mais pontos distintos. Estas redes permitem a comunicação entre pessoas (caso típico das redes de telefonia) ou a comunicação entre sistemas computacionais (como as redes de computadores que formam a Internet), através da transmissão de seqüências de bits (representando voz, vídeo ou dados) entre nós remotos.

Conceitualmente, uma rede de telecomunicações é composta por três planos ortogonais que contemplam diferentes aspectos da estrutura e da operação da mesma [ElSayed02]. O primeiro e principal plano é o plano de dados (ou de transporte), que é a parte da rede que transporta os dados dos usuários. Sua finalidade é bem clara uma vez que o principal objetivo de uma rede de telecomunicações é possibilitar a troca de informações entre seus usuários. Além

desta finalidade básica, funções como controle de fluxo e de erro também devem ser executadas neste plano. Para que uma rede seja capaz de funcionar corretamente, os nós devem trabalhar em conjunto para possibilitar que os dados cheguem aos seus devidos destinatários e com os níveis de serviço pré-determinados. Para tanto, informações de controle devem ser trocadas entre os nós que compõem a rede. O segundo plano, o plano de controle, ou de sinalização, é a parte da rede que lida com estas informações (ou sinais) de controle. Estes sinais permitem, por exemplo, o estabelecimento de chamadas em redes de telefonia, a criação de circuitos virtuais para a transmissão de dados em redes ATM [I.321] e a troca de informações de roteamento em redes IP [RFC791]. Há protocolos específicos de controle (ou sinalização) como o SS7 [Q.700] para telefonia, o PNNI [PNNI] para redes ATM e os protocolos BGP [RFC4271] e OSPF [RFC2328] para redes IP. O terceiro e último plano é o plano de gerência, que é a parte da rede que lida com aspectos administrativos e de manutenção dos equipamentos e serviços da rede. Funções como gerência de falhas, configuração de equipamentos, bilhetagem, desempenho e segurança são desempenhadas neste plano da rede. Alguns protocolos de gerência comumente utilizados incluem SNMP [RFC1157], CMIP [X.711] e TL1 [GR-831].

As principais tarefas tipicamente desempenhadas por sistemas de gerência são sintetizadas no termo OAM&P (*Operations, Administration, Maintenance, and Provisioning*). Operação diz respeito a atividades rotineiras de monitoramento do ambiente de rede, visando fundamentalmente à detecção, ao diagnóstico e à correção de falhas que porventura venham a ocorrer. São estas as atividades que mantêm a rede em funcionamento no curto prazo. A atividade de administração envolve fundamentalmente o planejamento da rede no longo prazo. Dados estatísticos sobre a operação e o uso da rede, tendências de mercado e estratégias organizacionais são levantados para posterior análise. Estes estudos e as decorrentes propostas de melhorias são fundamentais para que a rede permaneça confiável e adequada aos serviços que ela deve prover a seus usuários. As atividades de manutenção englobam a atualização, correção, ou substituição de equipamentos, criação e restauração de *backups*, entre outros. Normalmente são tarefas críticas que interrompem temporariamente o funcionamento da rede. Desta forma, a execução de tais tarefas deve ser planejada cuidadosamente para que causem o menor impacto possível aos usuários da rede. Por fim, as atividades de provisionamento dizem respeito à criação e ao estabelecimento de serviços de rede a um usuário. A remoção de tais serviços também é considerada uma atividade de provisionamento. Estas atividades podem envolver tanto a instalação de novos equipamentos quanto apenas a configuração de parâmetros de serviço.

Desempenhar todas essas tarefas de gerência em redes de telecomunicações é uma tarefa complexa e deve ser preferencialmente conduzida com o auxílio de sistemas de software desenvolvidos especificamente com este propósito.

De modo geral, a literatura de gerência de redes levanta três formas distintas de se gerenciar uma rede qualquer, a saber, (i) protocolos de rede (de quaisquer camadas) podem incluir neles próprios informações de gerência, (ii) protocolos de gerência específicos para uma determinada camada (de rede) podem ser estabelecidos e (iii) mecanismos de gerência podem ser implementados como uma aplicação. Estes três casos são conhecidos como operação de camada (*layer operation*), gerência de camada (*layer management*) e gerência de sistemas (*systems management*), respectivamente [Raman98]. A alternativa mais comum, e adotada pela maioria dos modelos de gerência (incluindo todos os modelos avaliados neste trabalho), é a terceira.

O fato de a gerência ser, em sua essência, uma aplicação sobre a rede implica, entre outras coisas, que os protocolos de gerência estejam na camada de aplicação. Há algumas desvantagens na adoção desta estratégia. Em primeiro lugar, ela torna mais complexo o desenvolvimento de equipamentos gerenciáveis, uma vez que estes precisam implementar protocolos de todas camadas de rede, mesmo que estes operem em camadas inferiores. Este é o caso, por exemplo, de *bridges* e de roteadores que operam na segunda e terceira camadas do modelo OSI [ISO7498], respectivamente. Uma segunda desvantagem é que, em caso de colapsos de rede com o comprometimento do funcionamento de camadas no nível de aplicação, as aplicações de gerência serão afetadas e ficarão incapazes de lidar com a crise. No entanto, apesar destas desvantagens, esta tem sido a solução preferida pela indústria e academia. Uma importante motivação para esta terceira estratégia é que aplicações de gerência geralmente são complexas, e o uso dos serviços prestados por camadas de rede intermediárias torna mais simples a implementação dos protocolos de gerência.

2.1.2
Padrões de Gerência de Redes

Introdução

Para suprir a demanda por gerência de redes de telecomunicações, os próprios fabricantes de equipamentos se encarregaram de desenvolver sistemas de gerência. Os primeiros sistemas desenvolvidos eram, entretanto, específicos para os equipamentos de um único fabricante, e seguiam modelos e protocolos de gerência proprietários. Este fato ocasionou dificuldades de interoperabilidade entre sistemas e equipamentos de diferentes fabricantes que interagiam em uma mesma

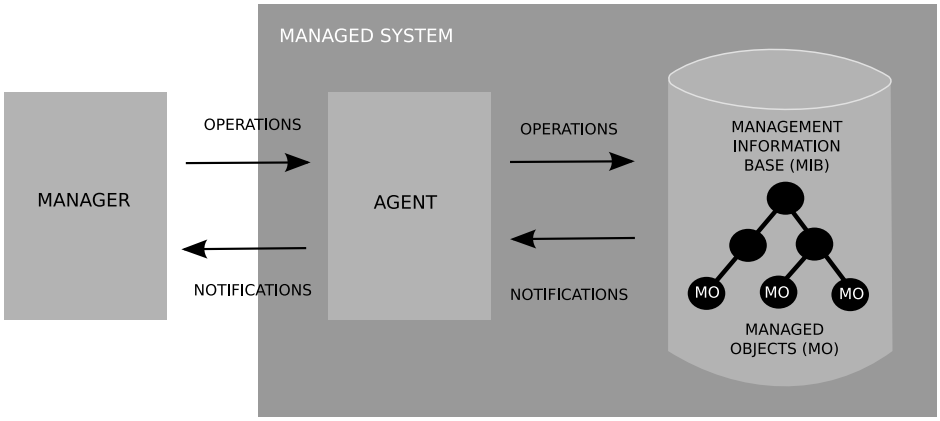

Figura 2.1: Paradigma Gerente/Agente

rede, causando grandes transtornos para os operadores da rede. Por conta disto, esforços de padronização de gerência de redes foram empreendidos por órgãos internacionais visando sanar tais questões. Estes esforços de padronização e suas principais contribuições serão abordados a seguir.

Um dos resultados mais importantes destes esforços, e que está fortemente consolidado no meio de gerência de redes, foi a arquitetura de gerência baseada no paradigma gerente/agente. De acordo com esta arquitetura, os recursos gerenciáveis de um equipamento (como por exemplo portas ou cartões) são modelados através dos chamados *objetos gerenciados*. Tais objetos gerenciados encapsulam as propriedades e funcionalidades de um recurso gerenciável e fornecem uma interface de acesso ao mesmo. Neste contexto, o agente pode ser entendido como o processo que possibilita que entidades externas acessem as propriedades e funcionalidades do conjunto de objetos gerenciados de um equipamento. O gerente, por sua vez, é a aplicação que acessa os objetos gerenciados expostos pelos agentes. O gerente tem como principal objetivo monitorar os recursos gerenciáveis e ajustá-los de acordo com os interesses dos administradores da rede. A Figura 2.1 ilustra o paradigma gerente/agente.

Cabe neste momento uma breve colocação acerca do termo “agente”. O termo “agente” tem sido amplamente utilizado pela comunidade de ciência da computação para descrever uma entidade de software qualquer que desempenha alguma tarefa de forma automática. Como esta descrição é um tanto genérica, grupos distintos desta grande comunidade acabaram atribuindo ao termo “agente” conotações distintas, trazendo, em algumas circunstâncias, dificuldades à comunicação entre os mesmos. Este conflito de nomenclatura é particularmente flagrante entre o grupo de gerência de redes e os grupos de inteligência artificial e de engenharia de software. Na literatura de gerência de redes, o termo agente é freqüentemente

usado para descrever o software embutido em equipamentos gerenciáveis (tais como roteadores ou *switches*) que responde a requisições feitas por estações de gerência e que envia alarmes à mesma [Yemini93, Stallings98a]. Este modelo de "agente" discutido no parágrafo anterior segue um paradigma cliente-servidor e é utilizado pelos mais importantes padrões de gerência de redes em uso na indústria apresentados a seguir. Por outro lado, a literatura de inteligência artificial e, mais recentemente, a de engenharia de software aplicam o termo "agente" para designar programas independentes capazes de agir de forma autônoma que buscam atingir metas pré-estabelecidas [Wooldridge95]. Este trabalho emprega o termo "agente" de acordo (primariamente) com a perspectiva da comunidade de engenharia de software. Entretanto, na presente seção (em que se abordam os padrões mais importantes de gerência de redes) este termo será freqüentemente empregado de acordo com a perspectiva da comunidade de gerência de redes. Espera-se que o contexto em que o termo estiver presente seja suficiente para dirimir eventuais ambigüidades.

Padrões de Gerência de Redes

Há algumas organizações que, ao longo dos anos, desenvolveram serviços, protocolos e arquiteturas para gerência de redes. Algumas delas são responsáveis pelos mais importantes padrões de gerência de redes utilizados pelo mercado. Três das mais importantes são citadas a seguir [Pras95]:

- International Organization for Standardization (ISO);
- International Telecommunication Union (ITU);
- Internet Engineering Task Force (IETF).

A ISO e a ITU, por exemplo, são autoras do importante modelo de referência OSI [ISO7498] de sete camadas, que engloba algumas das mais importantes abstrações usadas no ensino de conceitos de rede. Outras importantes contribuições destas organizações foram a pilha de protocolos que implementa o modelo de referência, assim como o protocolo de gerência CMIP [X.711].

Uma das mais importantes contribuições do ITU nesta área foi o TMN [M.3010], que é um modelo de gerência aplicável a redes de telecomunicações. Este modelo incorpora e usa muitos dos conceitos estabelecidos pelo modelo OSI e, embora não seja comumente aplicado em redes de dados IP (baseadas em comutação de pacotes), é utilizado freqüentemente para gerência de redes SDH [G.707] e GSM [Rahnema93], por exemplo [Tanaka98, Towle95].

A IETF, por sua vez, é responsável pela criação de muitos dos padrões da Internet, dentre os quais se encontram a pilha de protocolos TCP/IP [RFC1180]

e, em particular, o protocolo SNMP [RFC1157] utilizado para gerência de equipamentos de rede. Em alguns aspectos, os padrões da IETF se sobrepõem a padrões da ISO e da ITU. Esta sobreposição é bem nítida, por exemplo, quando se compara a pilha de protocolos TCP/IP (IP, OSPF, TCP e SNMP, entre outros) com a pilha de protocolos OSI (CNLP, IS-IS, TP4 e CMIP, entre outros). Detalhes sobre estas duas pilhas de protocolos podem ser encontrados em [Cisco03]. As principais razões que levaram a IETF e a ISO/ITU a seguir estas diferentes direções foram a demora para a conclusão dos padrões e protocolos OSI e, posteriormente, a relativa complexidade dos mesmos. Estes fatores levaram a IETF a desenvolver padrões próprios mais simples que os da OSI/ITU [Pras95].

A seguir, alguns aspectos importantes dos modelos e protocolos de gerência criados por essas organizações serão abordados.

Modelo de Gerência OSI

Por volta do início da década de 1980, soluções de redes de dados comumente utilizadas (como a SNA da IBM [Cisco03] e a DECnet da DEC [Cisco03]) eram baseadas em tecnologias proprietárias que não se integravam facilmente umas com as outras. Esta característica muitas vezes impactava negativamente a capacidade de usuários resolverem seus problemas através da integração de suas redes e sistemas. Como resposta a esta importante deficiência, a ISO (juntamente com a ITU) desenvolveu o modelo de uma rede padronizada e não-proprietária. Tal modelo, conhecido como modelo OSI, tinha como objetivo eliminar os problemas da pouca interoperabilidade então vivenciada e, desta forma, trazer mais flexibilidade a seus usuários [Russell06].

Algumas das contribuições mais importantes deste trabalho da ISO e ITU foram a criação do modelo de redes de sete camadas, a especificação de protocolos pertencentes a cada uma destas camadas, e, particularmente, o desenvolvimento do modelo de gerência OSI que será discutido a seguir.

O modelo de gerência OSI provê três importantes contribuições para a gerência de redes. Estas contribuições são tanto práticas quanto teóricas e representam um marco importante nesta área. Estas três contribuições são (i) o conceito das áreas funcionais, (ii) modelos de informação para representar recursos de rede e (iii) protocolos para transferência de informações sobre gerência de redes [Raman98]. Estes três aspectos do modelo de gerência serão abordados a seguir com mais detalhes.

A primeira contribuição do modelo de gerência OSI diz respeito ao conceito das áreas funcionais. Cada área funcional representa um tipo de atividade que a gerência de uma rede de telecomunicações deve desempenhar. As cinco áreas

levantadas pela OSI são também conhecidas pelo termo "FCAPS", formado a partir das iniciais de cada área (em inglês) e são sucintamente apresentadas a seguir.

Gerência de falhas. Responsável pela detecção, isolamento, notificação de administradores e, na medida do possível, correção de falhas na rede;

Gerência de configuração. Responsável pelo registro e manutenção dos parâmetros de configuração dos elementos de rede, assim como informações sobre versões de hardware e de software de cada um deles;

Gerência de contabilidade. Responsável pelo registro do uso da rede por parte de seus usuários com objetivo de posterior cobrança ou regulação de uso da mesma;

Gerência de desempenho. Responsável pela medição e disponibilização de informações sobre aspectos de desempenho da rede. Estes dados são usados para garantir que a rede opere em conformidade com os níveis de qualidade de serviço acordados com seus usuários;

Gerência de segurança. Responsável por restringir o acesso à rede por parte de elementos não autorizados e impedir que seus usuários a utilizem de forma irregular, intencionalmente ou não.

A segunda grande contribuição do modelo de gerência OSI foram os modelos para representação dos recursos de rede e de informações relacionadas a estes recursos. Para tal modelagem foi adotado o paradigma de orientação a objetos. Segundo esta modelagem, cada *objeto gerenciado* representa uma visão de um determinado recurso da rede que pode ser coordenada por um sistema qualquer de gerência. Exemplos de recursos de rede que geralmente são representados como objetos gerenciados são: portas de linha, cartões (com ou sem portas) e nós (equipamentos como roteadores ou *switches*). O fato de que objetos gerenciados representam visões de seus respectivos recursos significa que apenas alguns aspectos destes recursos são expostos a sistemas de gerência.

Como é de se esperar, por se tratar de um modelo orientado a objetos, existe o conceito de *classes* de objetos gerenciados, assim como o de *instâncias* de objetos gerenciados. Estes conceitos são equivalentes aos conceitos de classes e instâncias de objetos, respectivamente, do paradigma de orientação a objetos. Além disso, cada objeto gerenciado possui *atributos* e *ações*, equivalentes a atributos e métodos, respectivamente, em orientação a objetos. Estes atributos e ações podem ser agrupados em *pacotes* para facilitar a reutilização de especificações. Por fim, cada objeto gerenciado pode estar associado a *comportamentos*, que descrevem de

forma geral o modo de atuação do objeto, assim como *notificações*, que especificam os alarmes que o objeto pode emitir.

A especificação destes objetos é feita através das linguagens de notação ASN.1 [X.680], que descreve a sintaxe das estruturas de dados usadas pelos objetos, e GDMO [X.722], que descreve a estrutura dos objetos gerenciados propriamente ditos.

A identificação de um objeto gerenciado em uma base de objetos correspondente a uma rede de telecomunicações deve ser feita de modo inequívoco e sem ambigüidades. Para tanto os objetos são organizados de modo hierárquico em uma estrutura em forma de árvore. Isto é possível, uma vez que objetos gerenciados podem conter referências a outros objetos gerenciados. Há, entretanto, a restrição de que o valor do atributo de identificação de cada objeto contido por um mesmo objeto pai seja único. Esta restrição e a organização em árvore permitem a identificação inequívoca de um objeto gerenciado na chamada *Management Information Tree* (MIT).

A terceira importante contribuição do modelo OSI diz respeito aos protocolos e serviços que permitem a troca de informações entre equipamentos de rede e sistemas de gerência. Os serviços de gerência providos aos usuários são especificados pelo *Common Management Information Service* (CMIS) [X.710]. O protocolo que implementa estes serviços é conhecido como *Common Management Information Protocol* (CMIP) [X.711]. Apesar de haver uma clara separação entre serviço e protocolo, o termo CMIP é comumente utilizado para referir-se a ambos.

A Tabela 2.1 indica os diversos serviços oferecidos pelo CMIP. Estes serviços podem ser divididos em duas classes distintas. A primeira engloba reportes enviados autonomamente peloos elementos gerenciados aos sistemas de gerência para avisá-los sobre eventos relevantes. Apenas o serviço M-EVENT-REPORT se enquadra nesta classe. A segunda classe engloba as requisições feitas aos elementos gerenciados oriundas dos sistemas de gerência e as respectivas respostas. Todos os demais serviços se enquadram nesta classe.

Através destes serviços é possível ter acesso aos objetos gerenciados, tanto aos seus atributos quanto aos seus métodos (ações), bastando-se para tal, conhecer a posição do objeto na MIT. Há ainda dois mecanismos que permitem ao usuário do CMIP um maior controle sobre o universo de objetos envolvidos em uma determinada operação de gerência. Estes são os mecanismos de escopo (*scoping*) e filtragem (*filtering*) e estão disponíveis para os serviços M-GET, M-SET, M-ACTION e M-DELETE [Stallings93]. O mecanismo de escopo permite restringir o universo sobre o qual uma consulta ou ação referente a um serviço será aplicado. Esta restrição é feita determinando-se uma sub-árvore da MIT e a quantidade de níveis nesta sub-árvore sujeitos à ação do serviço. O mecanismo

Tabela 2.1: Serviços oferecidos pelo CMIP

Serviço	Descrição
M-EVENT-REPORT	Reporta um evento em um elemento gerenciado a um sistema de gerência
M-CREATE	Requisita a criação de uma instância de um objeto gerenciado
M-DELETE	Requisita a destruição de uma instância de um objeto gerenciado
M-GET	Requisita a busca de informação em um objeto gerenciado
M-SET	Requisita a modificação de informação em um objeto gerenciado
M-ACTION	Requisita que um elemento gerenciado execute uma ação
M-CANCEL-GET	Requisita que uma solicitação M-GET pendente seja cancelada

de filtragem também restringe o universo de objetos gerenciados sobre o qual um serviço CMIP será aplicado. Ele se dá através de um conjunto de expressões booleanas com assertivas sobre a presença ou o valor de um ou mais atributos dos objetos gerenciados. Apenas os objetos que satisfaçam estas expressões booleanas estarão sujeitos à operação de gerência em questão. Este mecanismo é semelhante, apesar de ser menos flexível, a uma cláusula WHERE em uma consulta SQL.

Modelo de Gerência TMN

Desde a década de 1980, a ITU tem trabalhado na especificação de um modelo genérico de gerência de redes de telecomunicações. A este modelo deu-se o nome *Telecommunications Management Network* (TMN) [M.3010]. Um dos propósitos deste modelo é ser aplicável em qualquer rede de telecomunicações, não importando as tecnologias de rede utilizadas ou o grau de heterogeneidade de fornecedores de equipamentos existente na mesma. Para tanto, foram definidos modelos e recomendações que permitissem, de forma padronizada, a operação, administração, manutenção e provisionamento (OAM&P) de redes e seus serviços nos complexos e heterogêneos ambientes tipicamente encontrados nas operadoras de telecomunicações.

O modelo TMN foi fortemente influenciado pelo modelo de gerência OSI, adotando alguns conceitos criados ou escolhidos por este. De fato, os dois modelos foram criados conjuntamente pela ISO e a ITU. Um exemplo da influência do modelo de gerência OSI sobre o padrão TMN foi a adoção do modelo gerente/agente e do paradigma de orientação a objetos.

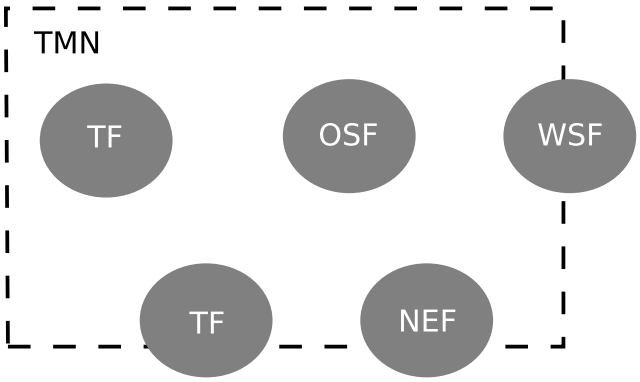

Figura 2.2: Blocos de função do TMN

Tabela 2.2: Blocos de função do TMN

Sigla	Nome	Descrição
OSF	Operation System Function block	Desempenha tarefas como monitoramento, coordenação e controle
NEF	Network Element Function block	Bloco de função correspondente a um elemento de rede gerenciado
WSF	Work Station Function block	Permite a interação de seres humanos com a TMN
TF	Transformation Function block	Conecta entidades funcionais com mecanismos de comunicação incompatíveis

De acordo com a Recomendação M.3010 da ITU-T, a TMN é, conceitualmente, uma rede de sistemas de gerência separada da rede de telecomunicações, mas que possui interfaces com os elementos de rede em pontos específicos. A arquitetura TMN definida nesta recomendação apresenta as entidades de uma rede TMN, assim como as comunicações entre estas, através de três diferentes perspectivas, a saber:

- Arquitetura funcional;
- Arquitetura física;
- Arquitetura de informação.

A arquitetura funcional define as principais funcionalidades sujeitas a padronização. Estas funcionalidades são mapeadas nos nós de uma TMN quando da implementação da mesma. Cada uma destas funcionalidades pode ser considerada uma entidade lógica da rede e recebe o nome de *bloco de função*. Há quatro tipos principais de blocos de função definidos no TMN, exibidos na Figura 2.2 e descritos na Tabela 2.2.

O bloco de função OSF é quem de fato desempenha as tarefas de gerência da rede. Ele monitora, coordenada e controla as funções de gerência da rede.

Tabela 2.3: Pontos de referência do TMN

Nome	Descrição
q	Relaciona OSFs a NEFs, a TFs e a outros OSFs. Também relaciona TFs a NEFs e a outros TFs.
x	Relaciona duas OSFs em TMNs diferentes.
f	Relaciona WSFs a OSFs e a TFs.
g	Relaciona WSFs a usuários humanos.
m	Relaciona TFs a equipamentos não-TMN.

Blocos OSFs podem estar conectados entre si, formando estruturas hierárquicas, por exemplo, para melhor desempenhar suas atividades. Também é possível conectar OSFs pertencentes a diferentes TMNs.

O bloco de função NEF apresenta a funcionalidade de um equipamento de telecomunicações, que é conduzir dados entre dois usuários da rede. Além disto, por ser o alvo principal de todas as atividades de gerência, este bloco deve possuir algum tipo de funcionalidade que permita que outros blocos de função, em particular o OSF, o gerenciem.

O bloco de função WSF provê os meios para que as informações de gerência sejam acessíveis aos administradores de redes, possibilitando que estes atuem manualmente na gerência da rede.

O bloco de função TF tem como finalidade possibilitar a conexão de duas entidades funcionais que apresentem mecanismos de comunicação incompatíveis entre si. Desta forma, ele age como uma espécie de *gateway* entre ambos. O bloco de função TF pode estar presente tanto no interior de uma TMN ligando dois outros blocos de função (padronizados mas com interfaces incompatíveis), quanto na fronteira de uma TMN servindo como canal de comunicação entre duas TMNs diferentes ou entre uma TMN e um ambiente não-TMN.

Os blocos de função interagem entre si através dos chamados *pontos de referência*. Os pontos de referência representam uma visão externa dos serviços oferecidos por um bloco de função. As ligações (ou interações) lógicas entre os blocos de função se dão através dos pontos de referência.

O TMN define cinco classes diferentes de pontos de referência, a saber: *q*, *x*, *f*, *g* e *m*. Estes pontos de referência permitem o relacionamento entre blocos de função conforme pode ser observado na Tabela 2.3. As classes q, x e f são descritas com algum detalhe na especificação. As classes g e m, entretanto, relacionam entidades externas à TMN e são descritas de forma parcial. A Figura 2.3 ilustra os blocos de função e os pontos de referência da arquitetura funcional do TMN.

Além da arquitetura funcional, o TMN também define uma arquitetura física. O propósito da arquitetura física é definir o mapeamento dos elementos da arquitetura funcional nos equipamentos físicos. A arquitetura física é composta

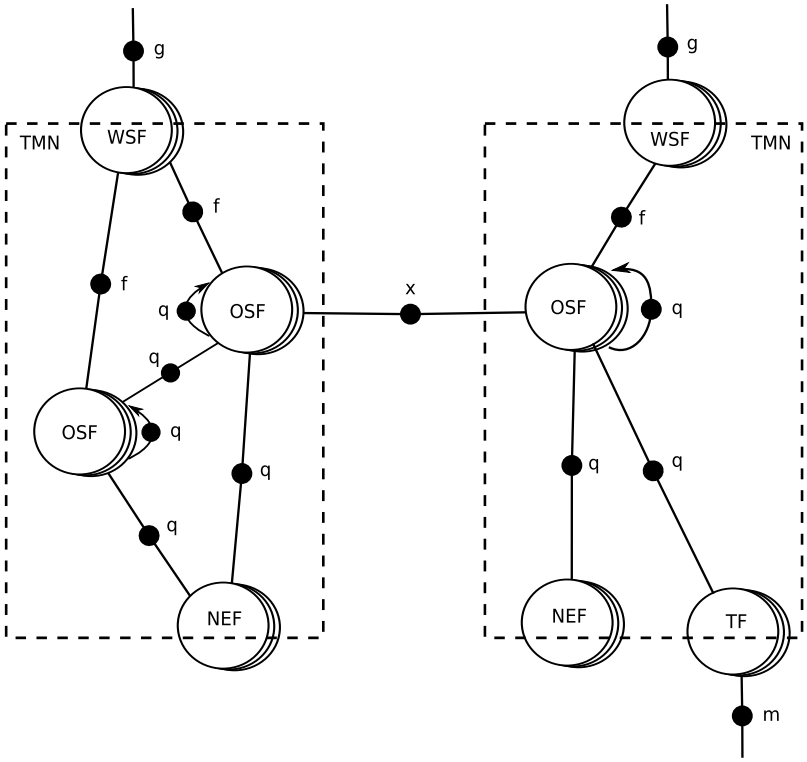

Figura 2.3: Arquitetura funcional TMN

de *blocos físicos* e *interfaces físicas*. Os blocos físicos representam os sistemas e equipamentos reais de uma rede de gerência que assumem a funcionalidade de um bloco de função da arquitetura funcional. Um bloco físico pode assumir a funcionalidade de mais de um bloco funcional. Os possíveis mapeamentos entre blocos de função e blocos físicos podem ser vistos na Tabela 2.4.

Como pode ser observado na Tabela 2.4, o padrão considera dois tipos diferentes de transformação, a saber, a adaptação e a mediação, que se aplicam aos pontos de referência q e x. Por esta razão há quatro blocos físicos relacionados ao bloco de função TF. A adaptação possibilita a comunicação entre um bloco físico TMN e uma entidade física não-TMN. A medição, por sua vez, permite a comunicação entre blocos físicos TMN que possuam mecanismos de comunicação incompatíveis entre si.

Há também um mapeamento direto entre pontos de referência e interfaces físicas. Este mapeamento está exposto na Tabela 2.5.

Como pode ser visto na Tabela 2.5, não há interfaces físicas definidas para os pontos de referência g e m. Isto se deve ao fato de que estes pontos de referência relacionam entidades externas ao TMN e, por conseqüência, a definição das interfaces associadas a estes pontos fica fora do escopo do padrão.

Tabela 2.4: Mapeamento entre blocos físicos e blocos de função

	NEF	TF	OSF	WSF
Network Element (NE)	M	O	O	O
Q-Adapter (QA), X-Adapter (XA), Q-Mediation (QM) e X-Mediation (XM)		M		
Operations System (OS)		O	M	O
Workstation (WS)				M

M: mapeamento mandatório; O: mapeamento opcional

Tabela 2.5: Mapeamento entre interfaces e pontos de referência

Interface física	Ponto de referência
Q	q
X	x
F	f

Apesar de o padrão TMN definir entidades como interfaces físicas, blocos físicos, pontos de referência e blocos de função, o grupo de entidades em que se concentraram os maiores esforços de padronização foi o das interfaces físicas. Pontos de referência e blocos de função são apenas abstrações e não há a necessidade de serem padronizados. Por outro lado, padronizar os blocos físicos poderia limitar ou dificultar desnecessariamente a implementação dos mesmos. Além do mais, a definição das interfaces físicas é suficiente para garantir a interoperabilidade entre os blocos físicos [Glitho95].

De todas as interfaces físicas, a que foi melhor definida foi a Q. Em versões anteriores do padrão, esta interface era denominada Q3. Esta antiga denominação ainda é bastante popular. De forma geral, o padrão define para as interfaces físicas (i) os protocolos de comunicação que podem ou devem ser usados, dentre os quais o mais relevante é o CMIP, (ii) diretrizes e recomendações relacionadas à documentação destas interfaces e (iii) diretrizes e recomendações para definição de novas classes de objetos gerenciados.

A terceira arquitetura definida pelo TMN é a arquitetura de informação. A TMN adota o modelo de representação baseado em objetos gerenciados e o modelo de comunicação gerente/agente usados pelo modelo de gerência OSI. Uma importante contribuição do padrão TMN neste terreno foi a definição de um conjunto de objetos gerenciados básicos usados para representar recursos de rede comuns (como equipamentos e circuitos) e também para desempenhar tarefas comuns (como envio de alarmes). Esta definição pode ser encontrada principalmente na Recomendação M.3100 [M.3100].

Por fim, uma última importante contribuição do modelo de gerência TMN foi a divisão das responsabilidades de sistemas de gerência de redes em quatro níveis

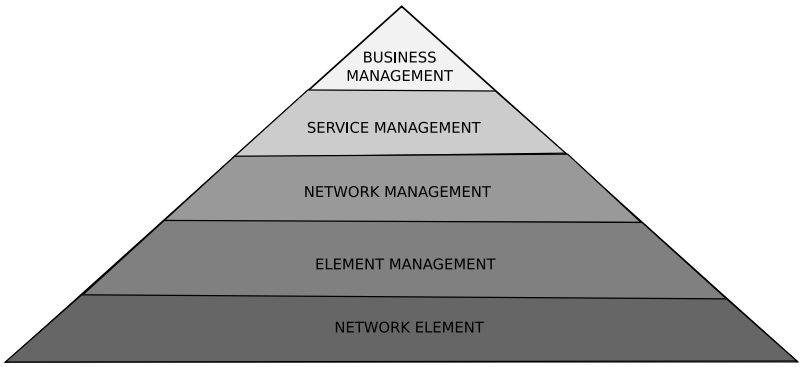

Figura 2.4: Camadas de gerência TMN

hierárquicos. Estes níveis são conhecidos como camadas de gerência e estão listados a seguir:

Camada de gerência de negócio. Esta camada tem um escopo bem amplo e engloba a gerência da organização como um todo. Um sistema de gerência pertencente a esta camada pode ser usado, por exemplo, como um suporte a processos de tomada de decisão relacionados a investimentos em recursos de rede. Dois outros possíveis usos seriam a gerência orçamentária e a gerência de recursos humanos relacionadas à gerência da rede;

Camada de gerência de serviço. Sistemas desta camada gerenciam aspectos relacionados aos serviços prestados aos clientes da organização, tais como configurações e provisionamento de serviços de comunicação. Um ponto importante monitorado nesta camada são os indicadores de qualidade de serviço (QoS);

Camada de gerência de rede. Sistemas desta camada gerenciam aspectos diretamente relacionados à interação entre equipamentos de rede, como por exemplo roteamento e congestionamento de enlaces;

Camada de gerência de elemento de rede. Sistemas desta camada gerenciam aspectos diretamente ligados a um equipamento de rede, tais como, coleta de estatísticas e detecção de erros.

A Figura 2.4 ilustra as camadas de gerência definidas pelo TMN. Há ainda uma quinta camada não gerencial que engloba os equipamentos de rede propriamente ditos.

Modelo de Gerência da Internet

O modelo de gerência da Internet como se conhece hoje surgiu a partir do final da década de 1980 em função da necessidade de se gerenciar redes IP, uma vez que estas já vinham apresentando considerável crescimento à época, o que dificultava seu gerenciamento. Neste período, os trabalhos de padronização de gerência de redes da ISO e da ITU ainda estavam longe de serem concluídos e não podiam ser aplicados em redes reais. Por tal razão, a IETF se empenhou em construir um modelo alternativo mais simples que pudesse ser concluído e implementado rapidamente. Na época, imaginou-se que este novo modelo seria apenas uma solução provisória para o problema de gerência de redes IP. O planejamento era que o modelo de gerência da Internet (baseado no protocolo SNMP) fosse eventualmente abandonado e substituído pelo modelo de gerência OSI (usando o protocolo CMIP) quando este fosse concluído [RFC1052]. Atualmente observa-se que isto acabou não ocorrendo: o modelo de gerência da Internet se tornou muito popular e o modelo OSI sobrevive apenas em nichos isolados.

O modelo de gerência de redes aplicado a redes IP possui quatro elementos principais [Stallings98a], a saber:

- Estação de gerência;
- Agente de gerência;
- Base de informações de gerência (MIB);
- Protocolo de gerência de redes.

A estação de gerência é o dispositivo em que se encontram as aplicações de gerência que efetuam tarefas tais como recuperação de falhas, configuração de equipamentos e análise de dados da operação da rede. É também através da estação de gerência que os operadores da rede podem monitorar e controlar a mesma. Para tanto, cada estação de gerência deve ser capaz de trocar informações de gerência com os equipamentos gerenciados.

O agente de gerência é o software presente nos equipamentos de rede que permite que estes sejam gerenciados remotamente. Este agente deve responder a requisições provenientes de estações de gerência solicitando informações, assim como enviar de forma pró-ativa alarmes a estações de gerência em caso de problemas.

As informações sobre os recursos da rede e o seu funcionamento devem ser estruturadas de forma padronizada para que os sistemas de gerência possam consultá-los e agir sobre os mesmos. Com este propósito foi definido o conceito da MIB, que nada mais é do que um conjunto de variáveis e tabelas que descrevem diversos aspectos do equipamento gerenciado. As MIBs são mantidas pelo agente

de gerência e cada equipamento de rede pode possuir uma ou mais MIBs de acordo com as funcionalidades e os protocolos que este implementa. Há diversas MIBs padronizadas pela IETF (assim como outras organizações) que consolidam as características mais comuns dos equipamentos de rede. Além disso, também é permitida a criação de MIBs proprietárias para a gerência das funcionalidades específicas de equipamentos que não sejam contempladas por MIBs padronizadas.

Por fim, existe o protocolo de gerência que permite que as informações de gerência sejam transmitidas entre agentes e gerentes. Este protocolo é o SNMP (*Simple Network Management Protocol*) [RFC1157], que é o elemento mais popular deste modelo de gerência. O SNMP apresenta três principais serviços: (i) *GET*, que possibilita a gerentes a obtenção de informações de agentes; (ii) *SET*, que permite que gerentes modifiquem parâmetros de configuração em agentes, e (iii) *TRAP*, que permite que agentes notifiquem sistemas de gerência acerca de problemas ou outros eventos relevantes. O serviço SET também pode ser usado para que um agente execute alguma ação sobre seu equipamento gerenciado. Esta ação normalmente é um efeito direto da mudança do valor de uma variável. Por exemplo, mudar o valor do status operacional de uma porta pode habilitá-la ou desabilitá-la. Tal comportamento deve ser documentado na especificação da MIB em que esta variável é definida.

Uma característica importante do protocolo SNMP é que ele normalmente opera acima do protocolo UDP [RFC768]. Isto faz com que as comunicações entre gerentes e agentes não sejam plenamente confiáveis, uma vez que os pacotes podem se perder ou chegar fora de ordem. As aplicações de gerência devem levar em conta esta característica e desenvolver seus próprios mecanismos para contorná-la. Um desafio particularmente difícil, entretanto, é garantir a confiabilidade no envio de TRAPs, uma vez que estes não são confirmados (o agente nunca recebe uma resposta indicando que seu alarme foi recebido com sucesso) e o sistema de gerência a princípio não tem ciência de que estes são enviados.

O protocolo SNMP é um componente central do modelo de gerência da Internet, e, ao longo de sua trajetória, três versões principais do mesmo foram definidas. As novas versões foram criadas basicamente para contornar deficiências da primeira versão. Segundo [Pras04], entretanto, é bastante improvável que haja uma nova versão do SNMP.

A primeira versão do SNMP (também conhecida como SNMPv1), definida em 1990 [RFC1157], foi fortemente inspirada nos conceitos de gerência de rede existentes na OSI antes da abordagem orientada a objetos ter sido escolhida. A idéia dos serviços GET e SET, o uso do paradigma gerente/agente e o uso da linguagem ASN.1, por exemplo, são comuns a ambos os modelos. Esta primeira versão do protocolo define cinco tipos de pacotes, a saber: (i) *GetRequest* e

(ii) *GetNextRequest*, para o serviço GET, (iii) *SetRequest*, para o serviço SET, (iv) *Response*, para respostas aos serviços GET e SET, e (v) *Trap*, para o serviço TRAP.

Um dos principais objetivos dos autores do SNMP foi criar um protocolo de gerência simples. Esta simplicidade acabou sendo um dos mais importantes fatores que contribuíram para a popularidade alcançada pelo SNMP. Por conta disto, entretanto, algumas funcionalidades foram deixadas de lado. Os três pontos mais criticados do SNMPv1 [Stallings98a] são a ineficiência no transporte de grandes blocos de dados, a inexistência de mecanismos de gerência descentralizada e a falta de segurança inerente ao protocolo.

A segunda versão do SNMP (SNMPv2), de 1993 [RFC1441], foi concebida com o intuito de solucionar os problemas listados acima. A ineficiência no transporte de grandes blocos de dados se devia basicamente a limitações dos pacotes GetRequest e GetNextRequest. Uma das limitações era que a transmissão de uma tabela (de uma MIB) devia ser feita linha a linha. Isto exigia diversas transações entre o gerente e o agente. Para resolver esta questão foi criado o pacote *GetBulk*, que permite a transmissão de mais de uma linha de uma tabela, reduzindo a quantidade de transações necessárias para a transmissão de tabelas.

Outra limitação da primeira versão do SNMP era que as transações GET eram atômicas. Isto significa que, por exemplo, quando uma solicitação GetRequest indica dois ou mais dados a serem buscados em uma MIB, e a busca de pelo menos um deles apresenta falha, nenhum dado é transmitido ao gerente. Ou o agente envia o conjunto completo de dados solicitados ou ele não envia dado algum, sinalizando que houve erro. Nestas situações de erro, o gerente deve diminuir o conjunto de dados sendo solicitados e efetuar uma nova transação. A melhoria da segunda versão em relação a esta questão foi o relaxamento da restrição de atomicidade das requisições GET. Com este relaxamento, todos os dados que puderem ser obtidos sem falhas serão enviados ao gerente, diminuindo a necessidade de novas transações.

Uma das críticas feitas ao SNMPv1 diz respeito à ausência de mecanismos que suportem uma arquitetura distribuída de gerência. O principal problema do modelo centralizado a que o SNMPv1 induz é que ele não escala bem na medida em que o número de agentes a ser gerenciado por um mesmo gerente aumenta. Em função disto, o SNMPv2 introduz um conceito de gerência distribuída, em que dois ou mais gerentes podem se organizar e dividir entre si a responsabilidade de controlar um grande conjunto de agentes. Duas entidades novas foram criadas para possibilitar a cooperação entre gerentes. A primeira é a MIB Gerente-Gerente [RFC1451], através da qual a cooperação entre gerentes pode ser conduzida. A comunicação entre os gerentes é feita através dos mecanismos comuns de GETs e

SETs, o que significa que um dos gerentes deve assumir o papel de agente. Outra nova entidade foi o pacote *Inform*, que permite que informações não solicitadas sejam enviadas de um gerente a outro, o que possibilita a notificação de eventos relevantes.

A especificação do SNMPv2 publicada em 1993 contemplava também mecanismos de segurança para transações SNMP. Esta proposta de segurança, entretanto, não foi muito bem recebida pela comunidade em função de uma falta de consenso com relação às estratégias adotadas e em função de ainda possuir deficiências [Stallings98b]. Por isto, foi publicada em 1996 uma nova revisão do SNMPv2 sem os aspectos de segurança. Esta versão revisada ficou conhecida como SNMPv2c.

Com as questões de segurança do SNMP ainda não resolvidas, novos trabalhos foram conduzidos e acabaram resultando na versão três do SNMP (SNMPv3) [RFC3411] primeiramente introduzida em 1998. O SNMPv3 inclui três importantes novos serviços, a saber, autenticação, privacidade e controle de acesso, que são brevemente discutidos a seguir.

O mecanismo de autenticação do SNMPv3 garante que uma mensagem recebida tenha sido de fato transmitida pela entidade cuja identificação consta no cabeçalho da mensagem. Também há a garantia de que esta mensagem não tenha sido alterada em trânsito nem artificialmente atrasada ou retransmitida. Este mecanismo de autenticação é baseado em chaves secretas previamente configuradas nas duas entidades (agente e gerente) que desejam se comunicar.

Para garantir a privacidade na troca de mensagens entre agentes e gerentes, é utilizado um mecanismo de criptografia baseado no DES [Menezes96]. Também é necessário que uma chave secreta seja previamente configurada em ambas as entidades. Este mecanismo de privacidade garante que nenhum terceiro seja capaz de decifrar mensagens SNMP eventualmente interceptadas.

O terceiro serviço de segurança introduzido nesta última versão do SNMP foi o controle de acesso. Com este serviço, um agente SNMP pode definir diferentes níveis de acesso à sua MIB, restringindo o acesso de gerentes à mesma. Há dois tipos principais de operações de limitação de acesso que podem ser impostas a gerentes. Com a primeira o agente pode limitar a parte da MIB visível a um determinado conjunto de gerentes. A segunda permite que o agente restrinja as operações SNMP que determinado conjunto de gerentes pode efetuar sobre sua MIB. Esta segunda limitação pode ser usada, por exemplo, para permitir apenas leituras na MIB (para tanto apenas o serviço GET seria liberado). Para que este serviço seja aproveitado, as políticas de segurança de cada gerente devem ser previamente configuradas no agente.

Outros Modelos (ou Protocolos) de Gerência de Redes

Os modelos e protocolos de gerência de redes apresentados nas seções anteriores podem ser considerados como os mais importantes em sua categoria. Isto se deve ao grau de influência que eles têm exercido na evolução da área de gerência de redes e também na grande aceitação e aplicação dos mesmos pela indústria nas redes que oferecem os serviços de telecomunicações utilizados pelas sociedades modernas. Entretanto, outros modelos ou padrões de gerência têm surgido ao longo do tempo e contribuído para o desenvolvimento de redes mais facilmente gerenciáveis. Alguns destes são citados a seguir.

O TINA-C (*Telecommunications Information Networking Architecture Consortium*) [Barr93] é um consórcio formado por empresas de telecomunicações no início da década de 1990. Ele tem como objetivo prover uma arquitetura baseada em tecnologias distribuídas (como CORBA [OMG04]) que possibilite a operadores de redes de telecomunicações a introdução rápida e flexível de novos serviços, assim como a gerência de serviços e da infra-estrutura de rede de forma integrada.

O RMON (*Remote Monitoring*) [RFC3577] é uma especificação de monitoramento que permite que *probes* localizadas em uma rede coletem dados estatísticos sobre a operação da mesma. Estes dados coletados são consolidados em uma MIB padronizada e podem ser transmitidos via SNMP a uma estação de gerência.

O WBEM (*Web-Based Enterprise Management*) [WBEM] é um conjunto de modernas tecnologias de gerência de sistemas desenvolvido para ambientes de rede, em especial redes de computadores corporativas. Estes padrões são desenvolvidos pela organização DMTF (*Distributed Management Task Force*). Fazem parte do WBEM tecnologias tais como o CIM (*Common Information Model*) [CIM], que é um modelo para descrever entidades comuns a sistemas de informação, e o WS-Management (*Web Services for Management*) [WSMAN], que permite a gerência de redes e aplicações através de Web Services [WS]. As tecnologias do WBEM têm sido adotadas por sistemas operacionais de diversos fornecedores. Exemplos desta adoção são o WMI (*Windows Management Instrumentation*) [Tunstall02] da Microsoft (Windows), o WBEM Services [WBEMServices] da Sun (Solaris) e o OpenWBEM [OpenWBEM] da Novell (SUSE Linux).

2.1.3 Paradigmas de Gerência

Com o avanço das tecnologias de redes de telecomunicações e das técnicas de gerência das mesmas, diferentes modelos e paradigmas de gerência de redes têm sido propostos e implementados. Um dos aspectos comuns aos modelos mais importantes apresentados na Seção 2.1.2 é a arquitetura baseada no paradigma

gerente/agente criada pela ISO e ITU e popularizada pelo SNMP. Em alguns casos, utilizam-se variações sofisticadas desta arquitetura que empregam vários gerentes organizados em diferentes níveis de gerência, responsáveis por tarefas específicas e que formam uma estrutura hierárquica de gerência de redes.

Não há no meio acadêmico um consenso universal sobre a classificação (ou mesmo sobre questões de nomenclatura) de modelos de sistemas de gerência, mas três modelos são comumente citados pela maioria dos pesquisadores, a saber: o centralizado, o hierárquico e o distribuído. Os modelos hierárquico e distribuído também são conhecidos como fracamente distribuído e fortemente distribuído, respectivamente. Um estudo mais completo sobre esta questão pode ser encontrado em [MartinFlatin99]. Neste trabalho são apresentadas duas taxonomias para classificação de sistemas de gerência de rede. A mais simples estabelece quatro principais paradigmas utilizados por sistemas de gerência de redes, a saber:

- Paradigma centralizado;
- Paradigma hierárquico fracamente distribuído;
- Paradigma hierárquico fortemente distribuído;
- Paradigma cooperativo fortemente distribuído.

O fator chave para esta classificação é o número de gerentes e de agentes presentes na rede, assim como características das relações entre os mesmos. Enquanto sistemas centralizados possuem apenas um gerente, sistemas distribuídos possuem mais de um. A diferença entre sistemas fracamente distribuídos e fortemente distribuídos é que, no caso daqueles, o número de gerentes é bem pequeno quando comparado ao número de agentes, normalmente não ultrapassando uma ou duas dezenas. A distinção entre sistemas hierárquicos e cooperativos, por sua vez, reside no tipo de interação entre gerentes e agentes, em particular no que diz respeito ao tipo de *delegação* que ocorre entre os mesmos. O conceito de delegação será examinado a seguir.

Em sistemas distribuídos, é comum que elementos que desempenhem uma atividade em conjunto solicitem uns aos outros a execução de determinadas tarefas. Isto pode ser feito, entre outras razões, para que as tarefas sejam desempenhadas pelos elementos mais apropriados (no caso de haver elementos especializados em um determinado tipo de tarefa) ou para que haja um balanceamento da carga de processamento entre os diversos elementos do sistema. Este processo de passagem da responsabilidade pela execução de uma tarefa de um elemento a outro recebe a denominação de *delegação* [MartinFlatin99]. Podem-se identificar dois tipos principais de delegação, a saber: a *delegação vertical* e a *delegação horizontal*. Quando a delegação ocorre entre um gerente e um agente, ou entre gerentes de

Tabela 2.6: Relação entre paradigmas e tecnologias de gerência

Paradigma	Tecnologia / Protocolo
Centralizado	SNMPv1
Hierárquico fracamente distribuído	SNMPv2 (com a MIB gerente-gerente) e OSI/TMN
Hierárquico fortemente distribuído	Código móvel e objetos distribuídos
Cooperativo fortemente distribuído	Agentes inteligentes

diferentes níveis hierárquicos, se tem a delegação vertical. Caso a delegação ocorra entre gerentes (ou mesmo agentes) de mesmo nível, ocorre a delegação horizontal. Assim sendo, seguem o paradigma hierárquico sistemas em que a delegação vertical é predominante sobre a horizontal. Caso o inverso seja observado, os sistemas em questão seguem o paradigma cooperativo.

Para a melhor compreensão destes paradigmas, a Tabela 2.6 exemplifica quais tecnologias (ou protocolos) implementam cada um dos paradigmas de gerência aqui expostos. As principais tecnologias representantes dos paradigmas centralizado e fracamente distribuído foram discutidas na Seção 2.1.2. As tecnologias dos paradigmas fortemente distribuídos serão apresentadas a seguir.

A arquitetura tradicional de gerência de redes, amplamente utilizada pela indústria nos dias de hoje, é fortemente baseada em soluções centralizadas (e também soluções hierárquicas fracamente distribuídas) de controle e gerência de serviços e equipamentos. Uma forte evidência deste fato é que os protocolos mais comuns para gerência de redes de telecomunicações ainda são o SNMP e o CMIP (que seguem os paradigmas centralizado e fracamente distribuído). A motivação para a criação destes protocolos foi resolver a questão da interoperabilidade entre diferentes fabricantes, e não houve, pelo menos inicialmente, uma preocupação em se adotar modelos de gerência distribuídos.

As principais críticas feitas ao modelo centralizado normalmente estão relacionadas a três características que são bastante desejáveis a sistemas de gerência, mas que são difíceis de se atingir devido às características de tal modelo. Estas três características são confiabilidade, escalabilidade e flexibilidade [Schönwälder00].

É difícil atingir-se níveis satisfatórios de confiabilidade em um sistema de gerência baseado no modelo centralizado uma vez que as estações de gerência representam pontos de falhas que podem comprometer todo o sistema se deixarem de operar corretamente. Outra situação em que há o comprometimento da gerência ocorre quando, devido a falhas em equipamentos, parte da rede se torna inalcançável a partir das estações de gerência. Em tais circunstâncias, um

subconjunto dos equipamentos da rede estaria fora do controle da gerência apesar de não haver problemas nas estações de gerência propriamente ditas. Obviamente, é extremamente desejável que redes de telecomunicações continuem em operação mesmo em caso de falhas de alguns de seus componentes.

O modelo centralizado enfrenta dificuldades em relação à escalabilidade por duas razões principais. A primeira é carga computacional exigida das estações de gerência que se tornam responsáveis por todos os equipamentos da rede. Outra é a carga de comunicações entre as estações de gerência e equipamentos. Como os recursos computacionais e de comunicação das estações de gerência são limitados, acaba criando-se um limite para o tamanho máximo da rede que estas estações podem suportar. Este problema é ainda mais grave em situações de falhas, quando se espera que a gerência tenha uma atuação mais intensa para contorná-las ou corrigi-las.

Outra questão é que, em um modelo de gerência centralizado, as funcionalidades de gerência são tipicamente pré-definidas e limitadas. Isto se deve ao fato destes modelos adotarem o paradigma cliente/servidor, em que há uma rígida associação de funcionalidades aos servidores durante a fase de projeto dos mesmos. Assim sendo, não há muitas possibilidades de se adicionar ou customizar funcionalidades de gerência, além, obviamente, do que o fabricante determinou. Esta pequena flexibilidade também se deve às limitações de recursos nas estações de gerência, uma vez que estas funcionalidades extras poderiam impactar ainda mais a escalabilidade do sistema de gerência.

A evolução natural do modelo de gerência centralizada foi o modelo de gerência hierárquico (também conhecido como modelo fracamente distribuído). A principal diferença entre os dois modelos é que não existe apenas uma única estação de gerência responsável por toda a rede. Há duas ou mais estações que dividem entre si a responsabilidade de administrar os diversos elementos e serviços da rede. Este modelo também introduz o conceito do gerente de gerentes. Em um modelo hierárquico, a rede é dividida em domínios e a cada domínio é atribuído um gerente. Cabe, então, a um gerente central controlar os gerentes de domínio. Nesta organização, o gerente central possui um nível de abstração mais elevado. Um exemplo bastante claro da evolução de um modelo centralizado para um modelo hierárquico pode ser observada entre a primeira e a segunda versão do SNMP, particularmente no que diz respeito à introdução da MIB Gerente-Gerente [RFC1451] e do novo serviço *Inform* que permitem a comunicação entre gerentes de diferentes níveis hierárquicos.

A grande vantagem do modelo hierárquico em relação ao modelo centralizado é a sua melhor escalabilidade, já que o limite para o tamanho máximo da rede é naturalmente aumentado devido à existência de várias estações de gerência.

Há ainda algum ganho em relação à confiabilidade do sistema, já que com este modelo não existe mais um ponto de falha único no sistema de gerência. Entretanto, a questão da falta de flexibilidade, apontada como uma deficiência do modelo centralizado, não é resolvida com o modelo hierárquico, uma vez que a relação entre agente e gerente não é alterada. Os agentes continuam se comportando apenas como meros coletores de dados.

Uma evolução ao modelo de gerência hierárquico é o modelo de gerência fortemente distribuído. A principal proposta deste modelo é permitir uma maior flexibilidade nas atividades de gerência de redes. Para que isto seja possível, os agentes passam a desempenhar papéis mais relevantes e ativos no sistema de gerência, não sendo apenas meros coletores de dados, mas também processando os mesmos. Isto contribui ainda mais para a descentralização do sistema de gerência, o que traz benefícios em relação à escalabilidade e à confiabilidade do sistema como um todo. Estudos quantitativos que indicam as vantagens do uso de modelos distribuídos podem ser encontrados em [Liotta99, Chen02]. A principal desvantagem de modelos distribuídos é que estes são bem mais complexos de se implementar que os modelos centralizados e fracamente distribuídos.

Há pelo menos três vertentes principais de sistemas de gerência distribuídos. Estas três vertentes são (i) código móvel, (ii) objetos distribuídos e (iii) agentes inteligentes (ou agentes de software) [MartinFlatin99].

Mobilidade de código é uma técnica para criação de sistemas distribuídos em que o código executável de um programa é transmitido através de uma rede e executado em diferentes nós da mesma. Um estudo detalhado do tema pode ser encontrado em [Fuggetta98]. Nele são identificadas as três principais formas de mobilidade de código, a saber, (i) *avaliação remota*, em que um cliente invoca um serviço ao servidor, passando-lhe não apenas o nome e os parâmetros de entrada do serviço, como também código do serviço para execução; (ii) *código sob demanda*, em que um cliente interessado em desempenhar alguma tarefa busca o código correspondente à tarefa em questão em um servidor e o executa; e (iii) *agente móvel*, em que uma unidade de execução (agente) migra autonomamente de nó a nó na rede, desempenhando sua atividade.

O trabalho que primeiramente explorou o uso de técnicas de código móvel aplicado à gerência de redes foi o de Goldszmidt *et al.* [Goldszmidt95], com o conceito de *Gerência por Delegação* (*Management by Delegation* – MbD). Este trabalho propõe uma extensão ao tradicional paradigma gerente/agente. A novidade proposta é que gerentes enviem a agentes programas para desempenhar tarefas de gerência. Estes programas são incorporados aos processos em execução nos agentes e executados juntamente com processos pré-existentes. Os resultados de tais programas são armazenados e posteriormente enviados aos gerentes. A principal

motivação desta técnica é a racionalizar o uso dos recursos da rede, evitando, por exemplo, que tabelas inteiras de uma MIB tenham que ser periodicamente transmitidas pela rede para a identificação de falhas.

As principais organizações de padronização na área de gerência de redes (ISO, ITU e IETF) trabalharam no sentido de integrar o modelo de Gerência por Delegação a seus padrões. Os principais resultados foram a Script MIB [Schönwälder00] desenvolvida para o modelo de gerência da Internet (SNMP) e o CMIP *Command Sequencer* [X.753] desenvolvido para o modelo OSI/TMN.

Uma segunda vertente de paradigmas hierárquicos fortemente distribuídos é a baseada em tecnologias de objetos distribuídos. Sistemas baseados em objetos distribuídos são sistemas que seguem os paradigmas de orientação a objetos, mas possuem a peculiaridade de que neles os objetos podem estar localizados em processos ou máquinas diferentes. Para que isto seja possível, há uma camada intermediária de software que provê serviços de serialização e transmissão de objetos entre processos ou máquinas, e serviços de descoberta de objetos remotos, entre outros. Algumas das principais tecnologias que permitem o uso de objetos distribuídos em sistemas são CORBA [OMG04], RMI [Sun] e DCOM [Microsoft].

Destas tecnologias de objetos distribuídos, a que alcançou a maior popularidade no meio de gerência de redes foi CORBA. É importante ressaltar que CORBA foi amplamente utilizada no contexto da TMN, particularmente na implementação de OSFs nas camadas de negócio e serviço (BML e SML) [Redlich98]. No entanto, este tipo mais comum de uso de CORBA em sistemas de gerência de redes não pode ser considerado como fortemente distribuído, uma vez que a relação entre gerentes e agentes permanece equivalente aos sistemas TMN tradicionais. O uso do CORBA só representa de fato uma evolução no sentido de um modelo fortemente distribuído quando os agentes são capazes de desempenhar papéis mais ativos na gerência da rede, garantindo maior flexibilidade ao sistema. Um exemplo de uso de CORBA em sistemas de gerência fortemente distribuídos pode ser encontrado em [Lazar97].

Houve um importante esforço no sentido de prover a interoperabilidade de sistemas legados baseados em CMIP e SNMP com a então emergente tecnologia CORBA. A força tarefa JIDM (*Joint Inter-Domain Management*) [JIDM] foi criada com objetivo de desenvolver os mecanismos para possibilitar esta interoperabilidade. Um dos principais resultados deste esforço foi o mapeamento dos modelos de informação dos modelos de gerência OSI e da IETF (baseados em ASN.1, GDMO e SMI) em estruturas de dados no formato CORBA IDL. Outro importante resultado foi a definição de interfaces CORBA IDL que implementassem todas as interações possíveis em CMIP e SNMP [JIDM00]. Estes dois resultados garantiram os mapeamentos estático e dinâmico (respectivamente) entre os modelos

OSI/TMN e IETF com a tecnologia CORBA e possibilitaram a interoperabilidade entre sistemas baseados nestes modelos de gerência.

Também é importante destacar a iniciativa da ITU e ISO de modernizar o padrão de gerência OSI/TMN através da adoção do paradigma de objetos distribuídos. O resultado destes esforços foi a ODMA (*Open Distributed Management Architecture*) [X.703]. No contexto da ODMA, não há mais gerentes e agentes com papéis fixos, o que caracteriza os paradigmas centralizado e fracamente distribuído. Pelo contrário, qualquer *objeto computacional* (entidade que substitui gerentes e agentes) pode possuir interfaces para gerenciar outros objetos computacionais – assumindo um papel de gerente – ou expor interfaces para permitir que ele seja gerenciado – assumindo um papel de agente.

Por fim, existem ainda sistemas de gerência baseados em um paradigma cooperativo, tipicamente utilizando agentes inteligentes (ou agentes de software). Neste contexto, o termo "agente" possui a conotação usada pela comunidade de inteligência artificial. O grande diferencial que o paradigma de agentes inteligentes apresenta em relação aos paradigmas de código móvel e objetos distribuídos é o fato que nele cada agente é pró-ativo no sentido de atingir determinada meta que lhe foi estabelecida. Esta característica torna mais natural a delegação horizontal de responsabilidades entre os agentes. As principais características de agentes inteligentes e sua aplicação no domínio de gerência de redes serão abordadas na Seção 2.2 e no Capítulo 5, respectivamente.

2.1.4 Tendências em Gerência de Redes

Durante muito tempo tem havido esforços para o desenvolvimento de técnicas e modelos para gerência de redes. As seções anteriores apresentaram alguns dos resultados mais relevantes e difundidos obtidos até o momento nesta área. Mas há também novas técnicas e modelos que representam as tendências para a gerência das novas gerações de redes. O uso de agentes de software é uma destas tendências. Outras duas tendências são sucintamente apresentadas a seguir.

Gerência Baseada em Políticas

O conceito central da gerência baseada em políticas [Strassner03] é utilizar um conjunto de regras (ou políticas) de alto nível para determinar o comportamento geral da rede. Obviamente deve haver mecanismos que mapeiem estas políticas abstratas nas respectivas ações práticas sobre os recursos de rede. A grande vantagem desta estratégia é permitir que operadores de rede especifiquem operações de gerência em termos dos objetivos que precisam ser alcançados, ao invés de ter

que descrever detalhadamente todas as operações que precisam ser executadas. Dois exemplos de trabalhos nesta área são [Yemini00, Verma02].

Há uma interessante relação entre gerência baseada em políticas e o uso de agentes de software para gerência de redes. Isto se deve ao fato de que políticas podem ser diretamente mapeadas em metas que agentes de software deliberativos devam alcançar. Isto permite um casamento entre gerência baseada em políticas e agentes de software. Esta estratégia é usada, por exemplo, em [Kohli03].

Gerência Baseada em XML

Outra tendência que tem ganhado força na área de gerência de redes é o uso de tecnologias baseadas em XML [XML]. Tal movimento tem sido identificado e discutido em trabalhos como [Schönwälder03, Pras04]. Podem-se citar algumas razões para este movimento. A principal razão é que as atuais tecnologias de gerência de redes não satisfazem todos os requisitos desejados por operadores de rede. Logo, novas alternativas estão sendo buscadas. Por outro lado, há uma significativa disseminação e uma ubíqua aplicação de tecnologias baseadas em XML no domínio de tecnologia de informação (TI) e na Internet. Muitos dos resultados destes esforços acabam sendo aplicáveis em domínios tais como o de gerência de redes.

Um ponto particularmente significativo nesta discussão é que algumas das características inerentes à tecnologia XML são bastante interessantes do ponto de vista de gerência de redes. Em [RFC3535] são apresentadas algumas vantagens do XML em relação às atuais tecnologias de gerência de rede, em particular o SNMP. Alguns dos pontos positivos do XML para gerência de redes salientados neste documento são (i) o fato de XML ser um formato de máquina facilmente processável e (ii) que conta com excelente suporte e disponibilidade de ferramentas e bibliotecas para sua manipulação. Além disso, (iii) a linguagem permite a modelagem de estruturas de dados com diferentes níveis de complexidade, e (iv) conta com excelentes mecanismos para definição de novos modelos de dados (DTD [DTD] e XSD [XSD]).

Um relevante exemplo de tecnologia XML aplicada à gerência de redes são os Web Services [WS]. Dois trabalhos que exploram este tema são [Pavlou04, Boutaba04]. Outros exemplos de aplicação de XML para gerência de redes são os padrões WBEM desenvolvidos pela DMTF mencionados anteriormente.

Por fim, uma promissora iniciativa da IETF foi o desenvolvimento do protocolo Netconf [RFC4741, Choi04], que é baseado em XML. Este protocolo tem sido indicado por alguns como um possível sucessor do protocolo SNMP. O Netconf permite não apenas o monitoramento do estado de equipamentos de rede, mas oferece também um mecanismo direto e flexível para a manipulação de

configurações de equipamentos de rede. A ausência de tais mecanismos é apontada como uma grave deficiência do SNMP. O Netconf foi influenciado por protocolos proprietários tais como o JUNOScript [Juniper].

2.2 Agentes de Software

2.2.1 Introdução

Com o desenvolvimento da ciência da computação, em particular a engenharia de software e as tecnologias da Internet, tem havido uma tendência à adoção de arquiteturas distribuídas e descentralizadas para o desenvolvimento de sistemas de software [Emmerich02, Papazoglou03]. Há desafios crescentes no que diz respeito à interoperabilidade de sistemas, à necessidade de operação em ambientes heterogêneos e ao percebido aumento de complexidade dos sistemas de hardware e software. Neste contexto, há quem diga que os modelos mais tradicionais de desenvolvimento de software, em particular as técnicas de orientação a objetos, têm limitações e que novos paradigmas devem ser buscados para enfrentar estes desafios [Zambonelli03]. Um paradigma relativamente novo que tem obtido destaque nos meios acadêmico e industrial e que tem potencial para resolver alguns destes problemas é a engenharia de software baseada em agentes de software [Luck04].

Nesta seção, alguns importantes aspectos acerca de agentes de software serão discutidos. Serão abordadas as características que distinguem agentes de software de outras abstrações existentes na área de engenharia de software, assim como características e arquiteturas de Sistemas Multi-Agentes (SMA). Por fim, serão abordados alguns padrões existentes para a implementação de SMA, um dos quais é empregado neste trabalho.

2.2.2 Características de Agentes de Software

Um agente de software pode ser entendido como um programa independente, que é capaz de agir de forma *autônoma* com o objetivo de atingir metas pré-estabelecidas. O conceito de autonomia neste contexto diz respeito ao fato de que um agente deve ser capaz de agir sem a intervenção direta de seres humanos ou mesmo de outros agentes. Além disso, ele deve ter controle direto e exclusivo sobre seu estado e comportamento.

Este modelo é particularmente diferente, por exemplo, do modelo de orientação a objetos, em que também há um encapsulamento de estado e de comportamento por parte dos objetos. Pode-se dizer que um objeto tem controle

de seu estado, uma vez que suas propriedades não são visíveis nem acessíveis a entidades externas. Isto também vale para agentes. O comportamento de objetos também é de certo modo controlado por estes uma vez que as possíveis mudanças de estado por que eles podem passar e as eventuais passagens de mensagens a outros objetos que eles podem vir a executar estão diretamente descritas e mapeadas em seus métodos. O que um objeto não controla é o momento em que seus métodos são invocados e executados. Sob este ponto de vista, um objeto não tem controle completo de seu comportamento. Seus métodos são invocados com o objetivo de atingir as metas das entidades que os chamam. Isto não ocorre com agentes, uma vez que suas ações não estão diretamente sujeitas à vontade ou influência de entidades externas. Por conta disto, pode-se dizer que, enquanto um objeto é passivo, um agente de software é ativo.

Outra característica interessante sobre agentes de software, e que é abordada na discussão sobre agentes feita por Wooldridge em [Wooldridge02], é o fato de que um agente deve estar sempre situado em um ambiente, que pode ser tanto real quanto virtual. Esta noção da presença de um agente em um ambiente é importante uma vez que remete para o comportamento *reativo* comum a agentes de software. O ambiente influencia o agente no sentido de que este age em função do que percebe ou observa do ambiente em que se encontra. Além disso, suas ações têm o objetivo de influenciar ou alterar o ambiente para que suas metas sejam alcançadas.

As questões da autonomia e reatividade são aspectos chaves na definição do que é um agente de software. Todavia, há diversos exemplos de sistemas autônomos e reativos que geralmente não são associados a agentes de software. Enquadram-se neste grupo os sistemas de controle que monitoram um ambiente do mundo real e que desempenham alguma ação para modificá-lo quando necessário. São representantes deste grupo os sistemas para automação predial e os sistemas de controle de vôo de aeronaves, por exemplo. Outro tipo de sistema autônomo e reativo é aquele em que se encontram os processos *daemon* de um computador. Tais processos monitoram um ambiente de software e também desempenham ações quando este sofre alguma alteração crítica. Um exemplo de processo *daemon* particularmente comum em gerência de redes são os agentes SNMP discutidos na Seção 2.1.

Esta constatação indica que há outros aspectos além da autonomia e reatividade que definem a essência de um agente. Destes aspectos, o mais significativo é provavelmente a pró-atividade. Este aspecto está intimamente ligado à capacidade de um agente de tomar a iniciativa de alguma ação que o ajude a atingir suas metas. Idealmente, um agente de software não tem todo o seu comportamento pré-determinado de forma rígida pelo programador que o criou. Em agentes deliberativos, é comum haver um conjunto de planos pré-determinados

de onde os agentes podem selecionar e executar aqueles que, de acordo com sua percepção e avaliação, o levem em direção aos seus objetivos.

Outro aspecto que diferencia agentes de software dos sistemas autônomos tradicionais é a sociabilidade. Por sociabilidade, entende-se a capacidade de um agente interagir com outras entidades externas, sejam elas outros agentes ou seres humanos. Ao interagir com outras entidades um agente tem como objetivo tanto a busca por recursos que possam ajudá-lo em seus próprios interesses assim como a cooperação com demais entidades que buscam resolver seus próprios problemas.

A lista a seguir, adaptada de [Wooldridge95], sumariza os mais importantes atributos de agentes de software aqui discutidos, e que os distinguem de outros paradigmas existentes na engenharia de software.

Autonomia: agentes de software operam com um mínimo de intervenção externa, seja de outros agentes ou de seres humanos, e, além disso, mantêm controle sobre o seu próprio estado e comportamento;

Sociabilidade: agentes de software podem se comunicar com outros agentes ou com seres humanos com a finalidade de trocarem informações ou cooperarem para atingirem objetivos comuns;

Reatividade: agentes de software percebem o ambiente em que operam e podem tomar ações em função de mudanças que venham a acontecer;

Pró-atividade: agentes de software devem agir orientados a metas, tomando a iniciativa através de planos e ações que os conduzam a seus objetivos.

Estes quatro atributos são freqüentemente considerados o mínimo necessário para que uma entidade de software seja considerada um agente. No referido trabalho estes atributos compõem o que os autores chamam de uma "fraca noção de agência". Não há, entretanto, um consenso universal sobre esta questão, visto que alguns autores excluem a pró-atividade da lista dos atributos essenciais de um agente. Desta forma, é possível definir entidades de software não deliberativas como agentes de software. Por outro lado, há atributos adicionais comumente encontrados e associados a agentes de software. Tais atributos podem ser desejáveis ou mesmo necessários em determinadas aplicações baseadas em agentes de software. A lista a seguir sintetiza alguns dos principais atributos adicionais de agentes de software [Franklin96].

Adaptabilidade: agentes de software podem traçar diferentes planos de ação para atingirem suas metas baseados nas condições do ambiente em que operam, e trocá-los em caso de mudanças no ambiente;

Aprendizagem: agentes de software podem possuir mecanismos de aprendizagem que os permitam adaptar-se a mudanças no ambiente em que atuam;

Mobilidade: agentes de software podem mudar dinamicamente o hospedeiro (*host*) em que executam, mantendo o seu estado, caso esta mudança seja benéfica para seus interesses.

2.2.3 Sistemas Multi-Agentes

Domínios tais como telecomunicações, gerência de processos de negócios, controle industrial, entre outros, são caracterizados pela complexidade dos sistemas neles existentes. A engenharia de software tem estudado e proposto métodos e procedimentos que permitam um melhor gerenciamento desta complexidade. Técnicas de análise e projeto de sistemas comumente aplicam a decomposição de problemas complexos em subproblemas menores, assim como a abstração do sistema através do uso de modelos simplificados e a organização dos diversos componentes do sistema de acordo com suas funcionalidades e dependências.

Muitos autores têm sugerido o uso de agentes de software como uma forma de lidar com tais questões. Neste contexto, os sistemas seriam compostos por diversos agentes, cada um com sua funcionalidade ou papel específico, formando os chamados sistemas multi-agentes (SMA). Cada agente de tais sistemas teria um ou mais objetivos próprios a alcançar e dependeria de outros agentes para tal. Idealmente, tais sistemas multi-agentes seriam capazes de convergir para os resultados esperados pelos usuários.

Segundo Jennings [Jennings01], a decomposição em agentes de software é um modo efetivo de particionar os problemas relativos a um sistema complexo. Ele também sustenta que as abstrações da chamada orientação a agentes são um meio natural de se modelar sistemas complexos. Por fim, ele defende que a filosofia de relacionamentos entre agentes é apropriada para se lidar com as dependências e interações que existem em sistemas complexos.

Alguns dos benefícios citados na literatura sobre o uso de sistemas multi-agentes são citados a seguir [Hayzelden99b]:

- Resolução de problemas que sejam grandes demais para uma única entidade (ou agente) centralizada;
- Redução de custos de processamento (é mais barato em termos de hardware usar vários processadores de desempenho mediano do que usar um único processador com um desempenho equivalente);
- Viabilização da interconexão e interoperabilidade entre sistemas legados existentes;

– Melhorias de escalabilidade, uma vez que a estrutura organizacional dos agentes pode ser alterada dinamicamente de forma a refletir mudanças em seu ambiente;
– Provisão de soluções distribuídas para problemas inerentemente distribuídos;
– Provisão de soluções para problemas cujas fontes de dados são distribuídas.

Além disso, sistemas que implementam o paradigma SMA são mais facilmente escaláveis devido a pelo menos três fatores. Um deles é a (i) modularidade que o paradigma naturalmente impõe, visto que cada agente pode ser entendido como um módulo do sistema que desempenha determinada funcionalidade. Outro fator é o (ii) fraco acoplamento entre seus componentes, uma vez que os agentes são entidades relativamente independentes entre si. Um último fator é o uso de (iii) abstrações de mais alto nível (o nível de abstração de um agente é maior que o de um objeto, por exemplo). Tais vantagens são extremamente convenientes para a construção de sistemas de gerência de redes como é o caso da arquitetura proposta neste trabalho. Estas vantagens são particularmente pronunciadas no que diz respeito à escalabilidade e à flexibilidade de tais sistemas.

2.2.4 Arquiteturas de Sistemas Multi-Agentes

Sistemas multi-agentes geralmente são considerados sistemas com um alto nível de complexidade devido ao fato de serem compostos por diversos componentes (agentes) que podem se relacionar – para efeitos práticos – de forma imprevisível ou indeterminada. Para lidar com sistemas complexos é comum o uso de modelos abstratos que permitam uma visão mais geral do sistema em questão. Tais modelos são muitas vezes denominados arquiteturas.

No caso de SMA, é possível identificar dois níveis de arquitetura. O primeiro diz respeito à modelagem dos componentes que formam a estrutura interna de um agente. Esta arquitetura pode ser denominada *intra-agente*. Um segundo nível de arquitetura diz respeito aos relacionamentos e interações entre agentes em um SMA. Esta arquitetura pode ser denominada *inter-agente* [Hayzelden99b].

Em [Wooldridge95] são identificados três tipos diferentes de arquiteturas intra-agentes, a saber, a arquitetura de agentes reativos, a arquitetura de agentes deliberativos e a arquitetura de agentes híbridos. Agentes reativos são aqueles que mantêm nenhuma ou pouca informação sobre o estado do ambiente em que se localizam. Estes agentes geralmente agem apenas em função de estímulos externos e seu comportamento é baseado em regras que vinculam uma determinada condição observada no ambiente a uma ação correspondente.

Agentes deliberativos, por sua vez, mantêm uma rica representação do estado do ambiente em que se encontram. Este conhecimento sobre o ambiente é utilizado

para o planejamento de ações futuras. Por estas razões, diz-se que estes agentes mantêm um *estado mental*. Estes agentes geralmente analisam diversas linhas de ação (ou planos) com o objetivo de identificar as ações que podem levar o ambiente para o estado que desejam. Estas características tornam estes agentes mais complexos que os agentes reativos, porém também os conferem maior flexibilidade. Um dos modelos mais utilizados para a implementação de agentes deliberativos é o modelo BDI (*Belief, Desire and Intention*) [Bratman87]. Este modelo é baseado nos conceitos de crenças, desejos e intenções. O conjunto de crenças representa o estado que o agente mantém sobre o ambiente. Desejos são os objetivos que o agente pretende eventualmente alcançar. Intenções são os desejos que o agente está comprometido em satisfazer através de planos em execução.

Por fim, agentes híbridos são aqueles que conjugam as características tanto de agentes reativos quanto de agentes deliberativos. Eles conjugam o mecanismo de resposta rápida baseado em estímulos externos assim como o comportamento pró-ativo possibilitado pela sua capacidade de planejamento.

No que diz respeito a arquiteturas inter-agentes, há pelo menos duas grandes vertentes, a de agentes cooperativos e a de agentes individualistas. Agentes cooperativos agem em conjunto para atingir um objetivo comum. Agentes individualistas agem em benefício próprio buscando satisfazer prioritariamente seus próprios objetivos. Um agente individualista somente coopera com outro agente caso receba algo de valor em troca.

Muitos esforços têm sido despendidos no estudo de agentes colaborativos. Algumas das principais razões citadas em [Hayzelden99b] para a distribuição de tarefas entre agentes colaborativos são: (i) a redução de custos de comunicação associados a uma entidade central, (ii) a melhoria de desempenho através de paralelismo de atividades, (iii) o ganho em reatividade devido a não necessidade de consulta a uma entidade central, e (iv) o ganho em confiabilidade devido a não dependência de um elemento central único.

A aplicação de agentes individualistas, por sua vez, está geralmente restrita a casos em que agentes que representam diferentes grupos ou organizações interagem entre si em ambientes abertos. É comum que esta interação se dê através de negociações em que um agente busca um serviço com o agente que lhe forneça as melhores condições. A implementação de SMA com agentes individualistas é mais difícil, pois há importantes aspectos de segurança que devem ser levados em consideração. Além disso, cada agente (ou o ambiente em que os agentes se encontram) deve estar preparado para lidar e se proteger de agentes "mal-intencionados".

2.2.5
Padrões de Sistemas Multi-Agentes

A tecnologia de sistemas multi-agentes (SMA) ainda é pouco utilizada fora do meio acadêmico e pode ser considerada relativamente imatura quando comparada com tecnologias mais tradicionais de engenharia de software. Isto é natural na medida em que ela é fruto de pesquisas recentes e ainda não teve alguns importantes problemas equacionados. Espera-se, entretanto, que esta tecnologia amadureça e se torne mais popular nos próximos anos [Nwana99a, Luck04].

Um fator que até pouco tempo dificultava uma adoção mais ampla da tecnologia de SMA era o fato de não haver padrões universalmente reconhecidos contendo especificações com os requisitos mínimos para a implementação de SMA e que garantissem a interoperabilidade entre diferentes agentes de software.

Para remediar esta situação importantes iniciativas no sentido de padronização têm sido tomadas. Uma das primeiras e mais influentes iniciativas foi a *Knowledge Sharing Effort* (KSE) iniciada por volta de 1990 pela DARPA [Neches91]. Seu objetivo era desenvolver técnicas, metodologias e ferramentas que permitissem o compartilhamento e a reutilização de conhecimento em sistemas de software. Apesar de esta iniciativa não ser focada exclusivamente em agentes de software, muitos de seus resultados foram aproveitados por esta comunidade. Um dos resultados mais importantes deste esforço, e que teve uma ampla utilização em SMA, foi a linguagem *Knowledge Query and Manipulation Language* (KQML) [Finin97].

A linguagem KQML permite a comunicação entre dois agentes de software de forma padronizada. Ela apresenta um alto nível de abstração, é orientada à troca de mensagens, e permite a troca de informações de forma independente da sintaxe ou ontologia usadas para representar o conteúdo das mensagens. A linguagem KQML pode ser dividida em três camadas, a saber, a camada de conteúdo, a camada de comunicação, e a camada de mensagem. A camada de conteúdo é a que traz o conteúdo da mensagem propriamente dito. A camada de comunicação contém parâmetros de nível básico da comunicação como, por exemplo, as identidades do remetente e destinatário e um identificador único relativo à comunicação. Por fim, a camada de mensagem determina o tipo de interação entre os agentes. Sua principal função é identificar o protocolo de rede a ser utilizado e o tipo da mensagem transmitida. A KQML é baseada em performativas da teoria de atos da fala [Searle69]. Isto permite que o remetente transmita sua intenção (informar, solicitar informação ou solicitar uma ação, por exemplo) quando do envio da mensagem [Labrou99].

A linguagem utilizada para comunicação entre dois agentes de software talvez seja o ponto mais crítico para permitir a interoperabilidade entre os mesmos.

Agentes que se comunicam precisam necessariamente de uma linguagem comum e mutuamente inteligível para que a troca de informações seja viabilizada. Além da linguagem de comunicação, também é necessário especificar e padronizar outros aspectos da comunicação entre agentes. Aspectos bastante pragmáticos como o registro da existência (e da saída) de um agente em um ambiente, e mecanismos para que se encontre um agente em particular neste ambiente (serviços de diretório e nome) devem ser padronizados para permitir a interoperabilidade entre sistemas. A padronização destas e de outras questões relativas à implementação de SMA tem sido o objetivo da organização FIPA (*Foundation for Intelligent Physical Agents*) desde a sua fundação em 1996.

Os padrões da FIPA não têm como objetivo determinar ou especificar a arquitetura interna dos agentes, mas sim prover interfaces que permitam a comunicação entre agentes. Os padrões da FIPA podem ser divididos em cinco grupos principais de acordo com seu assunto, a saber, arquitetura abstrata, transporte de mensagens, gerência de agentes, comunicação de agentes e aplicações de agentes [Dale01].

O propósito da arquitetura abstrata da FIPA [FIPA00001] é garantir interoperabilidade e reusabilidade. Para tanto são identificados os principais elementos necessários para a criação de SMA, assim como as relações entre estes elementos. A partir desta descrição abstrata de elementos e relacionamentos é possível construir implementações concretas e distintas de SMA que sejam compatíveis entre si uma vez que compartilham os mesmos conceitos e abstrações básicos.

O transporte de mensagens lida com o envio e a representação de mensagens através de diferentes protocolos de redes no contexto de uma plataforma de agentes. Para a FIPA, uma mensagem é composta por um envelope e um corpo. O envelope contém informações básicas (como remetente e destinatário) necessárias para que a plataforma entregue a mensagem de forma correta. O corpo da mensagem contém o conteúdo da mesma geralmente codificado na linguagem FIPA ACL [FIPA00061]. A FIPA padroniza o suporte para o serviço de envio de mensagens em uma plataforma de agentes [FIPA00067], diretrizes para o uso de protocolos tais como IIOP [FIPA00075], HTTP [FIPA00084] e WAP [FIPA00076], representações de envelopes de mensagens em XML [FIPA00085] e em formato binário [FIPA00088], assim como representações da linguagem FIPA ACL em formato texto [FIPA00070], XML [FIPA00071] e binário [FIPA00069]. Com estes padrões é possível obter-se uma interoperabilidade para a troca de mensagens entre agentes.

Outro importante padrão da FIPA é o que diz respeito à gerência de agentes. A especificação [FIPA00023] provê a arquitetura do ambiente em que

os agentes existem e operam. Nela são estabelecidos os modelos para criação, registro, localização, comunicação, migração e remoção de agentes. Dois dos mais importantes componentes aqui definidos são o serviço de diretório (*Directory Facilitator*) que permite o registro e a localização de serviços oferecidos por agentes e o sistema de gerência de agentes (*Agent Management System*) que controla os agentes em uma determinada plataforma assim como alguns aspectos do ciclo de vida dos mesmos.

Outro conjunto de padrões FIPA diz respeito à comunicação entre agentes. Ao contrário dos padrões relacionados ao serviço de transporte de mensagens, que lidavam com aspectos de baixo nível relacionados ao envio e representação das mensagens, os padrões de comunicação entre agentes definem um rico modelo semântico baseado na teoria de atos da fala [Searle69] para permitir comunicações de mais alto nível de abstração entre agentes. Através das chamadas performativas (tais como requisitar, confirmar ou informar) definidas em [FIPA00037], é possível que o agente que receba uma mensagem identifique melhor o contexto em que a mensagem deve ser interpretada. Outro padrão relacionado à comunicação entre agentes é a linguagem FIPA ACL [FIPA00061]. Esta linguagem provê mecanismos para adicionar contexto a uma mensagem, tais como a performativa (como indicado acima), remetente e destinatário e o protocolo de interação da mensagem. A linguagem FIPA ACL foi fortemente influenciada pela linguagem KQML apresentada anteriormente. Por fim, há ainda um conjunto de padrões que definem os chamados protocolos de interação, que nada mais são do que descrições de tipos de conversações comuns entre agentes. Dois exemplos de protocolos de interação são a interação de requisição [FIPA00026] em que um agente requisita que outro agente desempenhe alguma ação e a interação de consulta [FIPA00027] em que um agente solicita a outro agente a permissão para desempenhar alguma ação.

Por fim, o quinto e último conjunto de padrões especifica quatro aplicações baseadas em agentes. Também são descritos os serviços e as ontologias usadas em cada aplicação. Um dos casos especificados diz respeito ao provisionamento automático de serviços de rede para aplicações multi-mídia [FIPA00082]. Este exemplo é analisado com mais detalhes no Capítulo 5.