

Um Método de Resolução de Problema Reusável para Diagnóstico Automático no Domínio de Gerenciamento de Falhas em Redes de Comunicação

Marilza A. Lemos

Marcio Rillo

Universidade de São Paulo
Departamento de Engenharia Eletrônica
LSI - Laboratório de Sistemas Integráveis
Av. Prof. Luciano Gualberto, travessa 3, 158
Cidade Universitária - 05508-900 - São Paulo - SP
e-mail: {mlemos, rillo}@lsi.usp.br}

Resumo

Tarefas de gerenciamento de redes, tal como a tarefa de diagnóstico, têm sido tipicamente executadas por administradores de rede. Entretanto, as tarefas estão se tornando cada vez mais complexas devido ao crescimento da característica heterogênea e tamanho das redes atuais. Essas razões têm conduzido a pesquisa na área, em direção à automação de tarefas complexas em gerenciamento de redes. Nesse contexto, técnicas do campo da Inteligência Artificial (IA) têm sido utilizadas para suportar desenvolvimento de Sistemas Baseados em Conhecimento (SBC) no domínio. Na automação da tarefa de diagnóstico de falhas em redes, a principal abordagem adotada, tem sido a abordagem baseada em regras. Porém, sistemas baseados em regras não são flexíveis o suficiente para se ajustarem às mudanças que ocorrem nos elementos de rede e topologia das redes hoje, dificultando a manutenção e reuso desses sistemas. Na Engenharia do Conhecimento, uma área do campo da IA, a abordagem de modelagem do conhecimento permite a produção de descrições abstratas como componentes reusáveis na construção de SBCs, tais como: métodos de resolução de problemas (PSM), ontologias e modelos de domínio. Este trabalho utiliza tal abordagem para modelar um PSM reusável para a tarefa de diagnóstico no domínio de gerenciamento de falhas em redes de comunicação.

Abstract

Network management tasks, such as diagnosis task, have been typically performed by network administrators. However, these tasks are becoming more and more complex due to increase of the heterogeneous characteristic and size of networks at present. These reasons have led the research in this area towards to automation of complex tasks in network management. In this context, techniques of the Artificial Intelligence (AI) field has been used to support development of Knowledge Based Systems (KBS) in this domain. In the automation of faults diagnosis task in networks, the main approach adopted, has been the rule-based approach. But, rule-based systems are not enough flexible to adjust the changes that occur within the elements and topology of networks today, making difficult the maintenance and reuse these systems. In Knowledge Engineering, an area of the AI field, the knowledge modelling approach allows to produce abstract descriptions as reusable components for construction of the KBSs, such as: Problem-Solving Methods (PSM), domain ontologies and models. This work used that approach to model a reusable PSM to the diagnosis task in communication network fault management.

1 INTRODUÇÃO

Tarefas de gerenciamento de redes, tal como a tarefa de diagnóstico de falhas, têm sido tipicamente executadas por administradores de rede com o suporte de sistemas de gerenciamento (NMS - *Network Management Systems*). Tais sistemas interagem com elementos de rede (NEs - *Network Elements*) para obter dados sobre seu comportamento a fim de que ações possam ser tomadas pelo administrador para manter a rede num nível operacional aceitável. Um sistema de gerenciamento é uma aplicação distribuída apresentando duas entidades principais: o *gerente* e o *agente* [6]. O gerente é a entidade centralizadora capaz de enviar requisições de operações aos agentes que se encontram localizados nos diversos elementos de rede. Cada agente obtém as informações requisitadas acessando seus recursos gerenciados e enviam-nas ao gerente. O agente pode, ainda, enviar notificações assíncronas ao gerente. Para garantir interoperabilidade entre sistemas de gerenciamento, hardware e software

heterogêneos que constituem as redes, protocolos especiais foram desenvolvidos para a interação entre gerente e agentes, dos quais destacam-se o protocolo SNMP (*Simple Network Management Protocol*) para a arquitetura de redes Internet e o CMIP (*Common Management Information Protocol*) para a arquitetura de redes OSI (*Open Systems Interconnection*) [6]. Além disso, recursos de rede comuns às mais diversas configurações foram selecionados para compor conjuntos padronizados de recursos relevantes ao gerenciamento de redes. Notações padronizadas foram, também, estabelecidas para representar tais recursos, que assim modelados são denominados objetos gerenciados e encontram-se agrupados constituindo as chamadas MIBs (*Management Information Base*).

Apesar de todo trabalho já desenvolvido nessa área, resultado do esforço de grupos de trabalho inseridos em órgãos de padronização, administradores têm encontrado dificuldade para executar tarefas complexas de gerenciamento, tal como a tarefa de diagnóstico, devido ao grande número e diversidade de objetos e atributos que devem ser gerenciados nas redes atuais [13,14,18]. Essas razões têm conduzido a pesquisa na área, em direção à automação de tarefas complexas em gerenciamento de redes. Nesse contexto, técnicas do campo da Inteligência Artificial (IA) têm sido utilizadas para suportar desenvolvimento de Sistemas Baseados em Conhecimento (SBC) no domínio. Particularmente, na automação da tarefa de diagnóstico de falhas em redes, a principal abordagem adotada tem sido a abordagem baseada em regras. Porém, essa abordagem que define a tarefa de aquisição de conhecimento como a *transferência do conhecimento* de um especialista humano para dentro de um sistema automatizado, tem sido criticada por diversos pesquisadores [7,8,12,16] que apontam limitações tais como: a dependência entre regras e a existência de conhecimento de controle na suposta descrição declarativa da base de conhecimento (fatos e regras). Em consequência, o SBC é incapaz de explicar corretamente seu raciocínio, existe a dificuldade de inspeção, manutenção e reuso do sistema.

No domínio de gerenciamento de redes, as aplicações de gerenciamento fundamentadas sob a abordagem baseada em regras não são flexíveis o suficiente para se ajustarem às mudanças que ocorrem nos elementos de rede e topologia das redes na atualidade. Reusabilidade é um aspecto importante e que tem sido levado em conta na área de redes e seu gerenciamento devido à necessidade de interoperabilidade entre sistemas e equipamentos. As especificações padronizadas da área são descritas de uma maneira abstrata a fim de permitir que recursos de rede e funções (que manipulam tais recursos) sejam vistos da mesma maneira pelos projetistas de sistemas de gerenciamento, pelos fabricantes de equipamentos e administradores de rede. Visualizando a importância desses aspectos, também no que se refere às *aplicações de gerenciamento*, este trabalho procurou utilizar uma abordagem recente para construção de SBCs: a abordagem de modelagem do conhecimento. O uso de tal abordagem resulta num conjunto de componentes reusáveis tais como métodos de resolução de problemas (*Problem Solving Methods* - PSMs), ontologias e modelos de domínio, os quais são descritos no nível do conhecimento [12] e, portanto, independentes de formalismos de implementação. Esse resultado permite a reutilização dos modelos na construção de novas aplicações. Especificações que são reutilizáveis num domínio podem conduzir à especificações padronizadas quando o assunto se mostra de interesse em uma comunidade, característica desejável no mundo das redes de comunicação. Pesquisadores do campo da Inteligência Artificial estão, recentemente, descrevendo e reunindo componentes, modelados no nível do conhecimento, formando as chamadas *bibliotecas de modelos* [2,3,4,9,17]. Nelas, o engenheiro de conhecimento pode selecionar e adaptar modelos para a construção de sua aplicação.

De acordo com os problemas identificados no domínio de gerenciamento de redes e a abordagem escolhida no campo da Inteligência Artificial para resolver tais problemas, foram selecionadas duas ferramentas disponíveis na literatura de IA para especificar um *método de resolução de problema* reusável para a construção de SBCs que realizem a tarefa de diagnóstico no domínio de gerenciamento de falhas em redes de comunicação. O restante deste artigo está organizado do seguinte modo: a seção 2 descreve as duas ferramentas utilizadas na modelagem da tarefa diagnóstico para o domínio: a metodologia CommonKADS [4] e a Biblioteca de Benjamins [2]; a seção 3 estabelece uma representação para o domínio, baseada em modelo,

para manipulação pelo método e define ainda um conjunto de suposições para conduzir a seleção de métodos da biblioteca; a seção 4 apresenta o PSM proposto descrito através de três estruturas CommonKADS: a estrutura de tarefas e métodos, a estrutura de inferência e a estrutura de controle. Finalmente, a seção 5 apresenta as considerações finais deste artigo.

2 FERRAMENTAS PARA MODELAGEM DO CONHECIMENTO

2.1 A METODOLOGIA COMMONKADS

CommonKADS é uma metodologia para construção de SBCs fundamentada sob a abordagem de modelagem do conhecimento e que descreve comportamento de solução de problema, exigido ao SBC, através de quatro tipos de conhecimento: Conhecimento de Tarefa, Conhecimento de Inferência, Ontologia de Domínio e Modelo de Domínio.

Conhecimento de Tarefa

Em CommonKADS uma tarefa é caracterizada por: (i) um objetivo (*goal*) a ser alcançado, (ii) um método (PSM) capaz de alcançar esse objetivo e (iii) pelos dados de entrada e saída rotulados com nomes específicos da tarefa. Esses rótulos indicam os papéis que elementos reais do domínio desempenham no processo de resolução da tarefa (por ex. *hipótese*, *diagnóstico*). Por essa característica funcional eles são denominados *papéis de conhecimento* (*knowledge roles*). Um PSM é uma representação abstrata de um processo de raciocínio capaz de realizar uma tarefa. O PSM decompõe a tarefa em sub-tarefas e define um regime de controle entre elas. Podem existir várias maneiras de se atingir um objetivo e portanto vários métodos para a resolução de uma mesma tarefa. O conhecimento de tarefa é descrito através de três estruturas: estrutura de tarefas e métodos, estrutura de tarefas e estrutura de controle da tarefa.

Estrutura de tarefas e métodos (*task-method structure*) - Mostra graficamente a decomposição de tarefas e os possíveis PSMs que resolvem cada uma das (sub)tarefas e/ou inferências primitivas.

Estrutura de tarefas (*task structure*) - É uma árvore de tarefas e sub-tarefas onde a escolha dos PSMs já foi feita. Essa estrutura significa uma estratégia fixa para a solução de problema.

Estrutura de controle (*control structure*) - Descreve o regime de controle da tarefa, isto é, a descrição textual do *método de resolução* da tarefa. O regime de controle é expresso por construções de controle clássicas, tais como *loops* e iterações, descritas na forma de procedimentos envolvendo as sub-tarefas mínimas da tarefa, denominadas *inferências primitivas*.

Conhecimento de Inferência

O conhecimento de inferência é o conhecimento intermediário entre a estrutura de tarefas e a descrição do regime de controle da tarefa. Ele descreve o *fluxo de conhecimento* necessário para alcançar o objetivo da tarefa e é representado pela *estrutura de inferência* (*inference structure*). Essa estrutura é composta por (i) *inferências primitivas* que podem ser executadas sobre o conhecimento de domínio; (ii) *papéis de conhecimento* que elementos do domínio podem desempenhar na solução da tarefa e (iii) *conexões* entre as inferências primitivas através da dependência entre seus papéis de conhecimento de entrada e saída.

Ontologia do Domínio

Um PSM manipula uma descrição do domínio no qual se deseja executar a tarefa. CommonKADS descreve o conhecimento do domínio em dois níveis de abstração: o nível denominado *ontologia do domínio* e o chamado *modelo de domínio*. A ontologia de domínio é uma descrição da estrutura das declarações que podem ser feitas para descrever o domínio, constituindo-se uma meta-descrição do domínio. A ontologia de domínio tem duas funções: (1) fornecer um vocabulário para expressar modelos de domínio e (2) auxiliar a identificação de *papéis de conhecimento* do método (que resolve uma tarefa). O conjunto de *papéis de*

conhecimento de um método é denominado *ontologia da tarefa* quando a descrição desse método está num nível de abstração independente de domínio. Porém, quando a descrição de um método incorpora características de um domínio específico, o conjunto de *papéis do conhecimento* é também denominado *ontologia do domínio*. O conjunto de primitivas utilizadas para descrever ontologias de domínio em CommonKADS são: conceitos, propriedades, relações entre conceitos e relações entre expressões de propriedades. Exemplos de conceitos podem ser: *componente*, *observação*, *sintoma*, onde esses três conceitos podem constituir uma *ontologia de tarefa* para diagnóstico pois são suficientemente genéricos. Exemplos de conceitos que podem fazer parte da *ontologia de domínio* de gerenciamento de redes são: *elemento_de_rede*, *estado-remoto*. Exemplos de relações podem ser: relação *causa*, relação *part-of*. Exemplos de relações entre conceitos podem ser: *estado causa estado*, *componente part-of componente*.

Modelo de Domínio

Modelos de domínio são instâncias da ontologia de domínio que utilizam os termos reais do domínio. A ontologia de domínio serve como um gabarito (*template*) para descrever o conhecimento do domínio. Por exemplo, relações *is-a* instanciadas com elementos reais do domínio, descrevem modelos taxonômicos (hierarquias de sub-tipos), relações *part-of* instanciadas descrevem modelos estruturais. As relações entre expressões de propriedades descrevem modelos comportamentais do sistema. O modelo de domínio é manipulado pelo conhecimento de tarefa (o PSM) através dos papéis de conhecimento.

2.2 A BIBLIOTECA DE PSMS DE BENJAMINS

A consulta a uma biblioteca de modelos de conhecimento torna a atividade de aquisição de conhecimento menos complexa, pois reúne conhecimento da literatura estruturado de uma maneira reusável na construção de novos sistemas. A biblioteca de Benjamins [2] consiste num conjunto de métodos para diagnóstico, descritos no nível do conhecimento, que podem ser reutilizados na modelagem de tarefas de diagnóstico para diversos domínios. A tarefa de aquisição de conhecimento realizada neste trabalho está suportada por essa biblioteca, da qual métodos são selecionados e adaptados para compor o método proposto.

A biblioteca está organizada numa estrutura de decomposição de tarefas e métodos onde a tarefa de diagnóstico encontra-se decomposta em sub-tarefas e estas, por sua vez, em outras sub-tarefas. No último nível dessa hierarquia encontram-se sub-tarefas mínimas (inferências primitivas). Para cada sub-tarefa são propostos vários métodos que a resolve.

Dependendo das características do domínio no qual se quer executar uma tarefa, a escolha de um método pode ser mais adequada do que outra. A seleção de um método na biblioteca é guiada pelos *critérios de adequabilidade de métodos* que representam condições que um método exige para a sua utilização. Em relação ao domínio, tais condições descrevem um conjunto de restrições sobre como modelar o domínio para que seja possível aplicar o método desejado. Cada PSM da biblioteca é descrito pelas estruturas de conhecimento CommonKADS (*estrutura de tarefas e métodos*, *estrutura de controle*, *estrutura de inferência*), e pelos *critérios de adequabilidade*. Exemplos de critérios exigidos por métodos na biblioteca são: pontos alcançáveis para observabilidade do dispositivo, existência de determinado tipo de modelo (causal, funcional, estrutural, topológico), ferramentas para medição, banco de dados com valores esperados. A seção seguinte estabelece um conjunto de suposições para a seleção de métodos da biblioteca a fim de compor o PSM proposto.

3 REPRESENTANDO A REDE EM DIAGNÓSTICO BASEADO EM MODELO

Conforme mencionado anteriormente para que um método possa ser aplicado num domínio é necessário que esse domínio atenda alguns critérios estabelecidos pelo método [10]. No caso de métodos baseados em modelos para diagnóstico, um critério fundamental refere-se à existência de algum tipo de modelo que represente o comportamento do dispositivo a ser diagnosticado. O tipo de modelo disponível no domínio, implica no uso de um método capaz de raciocinar com

esse tipo em particular. Assim, pode-se afirmar que existe uma dependência entre os tipos de modelos disponíveis no domínio e os métodos que podem ser utilizados para resolver a tarefa nesse domínio. A escolha de um PSM requer uma análise dos modelos existentes, assim como a possibilidade e complexidade na construção de tais modelos num dado domínio. Os tipos de modelos clássicos que são propostos na literatura [1] para diagnóstico baseado em modelo são:

Modelos Estruturais - descrevem a estrutura física do dispositivo usando a relação *connected-to*. Seus constituintes básicos são componentes físicos e conexões entre os componentes.

Modelos part-of - descrevem como componentes consistem de outros (sub) componentes através da relação *part-of*. Em outras palavras, um modelo *part-of* captura os diferentes níveis de agregação existentes entre componentes.

Modelos Comportamentais - descrevem o comportamento de cada componente do modelo estrutural, através de seus estados internos ou valores de variáveis de seus terminais. Essa descrição é independente de qualquer configuração funcional com outros componentes.

Modelos Topológicos - descrevem a disposição (*layout*) dos componentes do dispositivo em termos de sua localização relativa ou absoluta. São úteis para cálculos de custo de obtenção de observações no dispositivo (custo de *probes*) e realização de reparo.

Modelos Funcionais - interpretam conhecimento comportamental de acordo com funções pretendidas pelo projetista. Eles contêm entidades tais como: funções, parâmetros de função, relações *part-of* entre funções em diferentes níveis de agregação, restrições entre parâmetros e cálculo que especifica como obter valores de parâmetros.

Modelos de Estados - esse modelo representa estados comportamentais relevantes do dispositivo e relações entre eles. As relações normalmente refletem causalidade (relação *causa*), isto é, a ocorrência de um estado causa a ocorrência de outros estados (efeitos) e, nesse contexto, são conhecidos como *modelos causais de estados*.

Modelos de Uso - refletem a percepção do dispositivo pela visão do usuário. São importantes para mapear os termos usados pelo usuário aos termos usados por outros modelos utilizados pelo raciocinador do SBC.

Redes de computadores são artefatos construídos pelo homem. Diferentemente de sistemas naturais como o corpo humano, a estrutura e funcionamento dos diversos componentes de uma rede de comunicação, suas dependências e relações são teoricamente e na prática conhecidas pelos seus projetistas e especialistas. A partir desse conhecimento é possível construir modelos de redes tais como os mencionados acima. Porém, a diversidade de componentes, a complexidade funcional e comportamental de uma rede de computadores é tão grande que arquiteturas especiais para gerenciamento foram estabelecidas tais como: arquitetura OSI e arquitetura SNMP. Tais arquiteturas reduzem a visão funcional e comportamental das redes de computadores abstraíndo detalhes de implementação e detalhes funcionais de protocolos de comunicação e componentes físicos. Os elementos da rede a serem gerenciados são modelados como objetos e denominados objetos gerenciados, pois são elementos que possuem observabilidade: podem ser lidos e, às vezes, alterados através dos protocolos de gerenciamento. Portanto, a escolha de um tipo de modelo para representar a rede, neste trabalho, baseia-se à visão que tais arquiteturas de gerenciamento oferecem.

Em estudo realizado em [11] foram identificados alguns tipos de modelos padronizados e estabelecidos pelos órgãos de padronização (ISO/OSI, IETF, ITU-T), tais como: modelos *part-of* e modelos taxonômicos. Os modelos *part-of* correspondem à hierarquia de *containment* que relaciona os recursos da rede (objetos gerenciados) numa hierarquia de decomposição de seus sub-componentes com seus respectivos nomes identificadores. Os modelos taxonômicos correspondem à definição de classes e subclasses de objetos gerenciados, compondo a hierarquia de herança das MIBs.

Apesar dos relacionamentos já modelados nessas hierarquias, a tarefa de diagnóstico necessita de modelos que retratem relacionamentos de causa e efeito, uma vez que a tarefa de diagnóstico consiste na busca de uma causa (falha) que gera um conjunto de efeitos (sintomas). No domínio de gerenciamento de redes, onde inúmeros indicadores de comportamento da rede já estão definidos nas MIBs, através dos atributos de objetos gerenciados, estados-falhos da rede

podem ser gerados através do processamento adequado desses indicadores. Portanto, a rede pode ser representada através de modelos causais de estados-falhos. Para a construção do PSM proposto neste trabalho foi assumida tal representação. A idéia de se construir o modelo de um dispositivo para a tarefa de diagnóstico reside no fato de que tal modelo será usado para comparações com o estado em que o dispositivo se apresenta num dado momento em que é observado. A figura 1 mostra um modelo causal de estados-falhos.

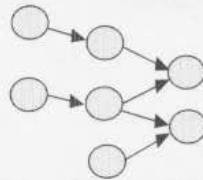


Figura 1 - Modelo causal onde cada círculo representa um estado-falho

3.1 SUPOSIÇÕES E RESTRIÇÕES SOBRE O DOMÍNIO

A escolha da representação da rede através de modelos causais de estados-falhos impõe alguns outros critérios ou suposições ao domínio. Esta seção enumera tais suposições a fim de guiar a seleção de métodos da biblioteca de Benjamins. Quando se pretende desenvolver um SBC para realizar uma determinada tarefa, é necessário identificar as dependências entre os vários tipos de tarefas encontradas nesse domínio [5]. Essa necessidade origina-se do fato de que a execução de uma tarefa pode depender da realização anterior de outro tipo de tarefa. No domínio de gerenciamento de redes, a tarefa de diagnóstico necessita que uma tarefa de *monitoração* seja realizada anteriormente a fim de obter e acumular dados a respeito dos recursos gerenciados. Uma outra aplicação de gerenciamento, denominada aqui como *gerador_de_eventos*, deve ser capaz de relacionar as informações monitoradas a fim de expressar a ocorrência de estados particulares da rede (eventos) que estão denominados neste trabalho como estados-falhos. Da mesma forma, para que uma tarefa automática de *reparo* possa ser executada é necessário que uma tarefa de diagnóstico seja, anteriormente executada, indicando o diagnóstico. Portanto, as dependências de entrada e saída dessas tarefas podem ser apresentadas pela série:

$$\text{monitoração} \rightarrow \text{gerador_de_eventos} \rightarrow \text{diagnóstico} \rightarrow \text{reparo}$$

Para construir o método para a tarefa de diagnóstico no domínio de redes, será assumida a existência de dados gerados por outras aplicações de gerenciamento, tais como aplicação de *monitoração* e *gerador_de_eventos*.

Outro critério que deve ser atendido para que o método proposto trabalhe é a existência do modelo topológico da rede. A introdução desse conhecimento num sistema de diagnóstico, permite dividir o espaço de estados-falhos da rede, de tal forma que o método pode percorrer o espaço de estados de maneira organizada, apoiada na topologia da rede.

A suposição seguinte refere-se ao número de falhas que podem ser manipuladas pelo método durante uma sessão de diagnóstico: falha única ou falhas múltiplas. Cada falha que ocorre num dispositivo gera um conjunto de sintomas. Um sistema de diagnóstico deve identificar a causa (falha) para cada conjunto de sintomas observados durante uma sessão de diagnóstico. O disparo de sessões de diagnósticos pode ser, tipicamente, controlado através de janelas de tempo, isto é, conjuntos de sintomas são analisados numa mesma sessão desde que tenham ocorrido no mesmo intervalo de tempo. Esse controle evita que falhas se acumulem gerando situações de falhas múltiplas. Falhas múltiplas são mais difíceis de serem identificadas e exigem métodos mais complexos e maior tempo de processamento do que a situação de falha única. O método proposto neste artigo assume a suposição de falha única para o domínio de gerenciamento de redes, uma vez que o domínio oferece facilidade na obtenção de informações temporais sobre a ocorrência de eventos na rede.

Uma última suposição feita ao domínio relaciona-se com as hipóteses geradas pelo método. Como a rede está representada por modelos causais entre estados-falhos, as hipóteses também o são. Portanto a dependência causal entre estados do modelo estabelece também a dependência entre hipóteses. Em resumo, as suposições e restrições estabelecidas no domínio de gerenciamento de redes para a construção do método proposto são:

1. ocorrências de estados relevantes da rede são registrados numa base de dados (denominada aqui como *log_eventos*) e encontra-se disponível para manipulação pelo método, a fim de detectar sintomas e obter informações adicionais para testar hipóteses. O *log_eventos* é o resultado da execução das aplicações de *monitoração* e *geradora de eventos*;
2. existência de modelos causais de estados falhos dos elementos de rede gerenciados;
3. existência do modelo topológico da rede gerenciada;
4. suposição de falha única;
5. hipóteses dependentes.

3.2 MODELANDO A ONTOLOGIA DO DOMÍNIO

A definição dos tipos de modelos para representar a rede permitem identificar as unidades desses modelos: *estados-falhos* (no modelo de estados) e *elementos de rede* (no modelo topológico) e caracterizá-las segundo o domínio e a tarefa a ser realizada (diagnóstico). Esse tipo de conhecimento, obtido num nível de abstração superior ao *modelo de domínio CommonKADS*, é uma meta-descrição dos modelos de rede e corresponde à *ontologia do domínio CommonKADS* conforme explicado na seção 2. Esse conhecimento é utilizado pelo método de duas maneiras: (1) para analisar e interpretar estados-falhos de um modelo de estados de maneira organizada; (2) para realizar a transferência de um modelo à outro durante a solução de problema [1]. A figura 2 mostra a ontologia de domínio proposta para o domínio de gerenciamento de falhas em redes.

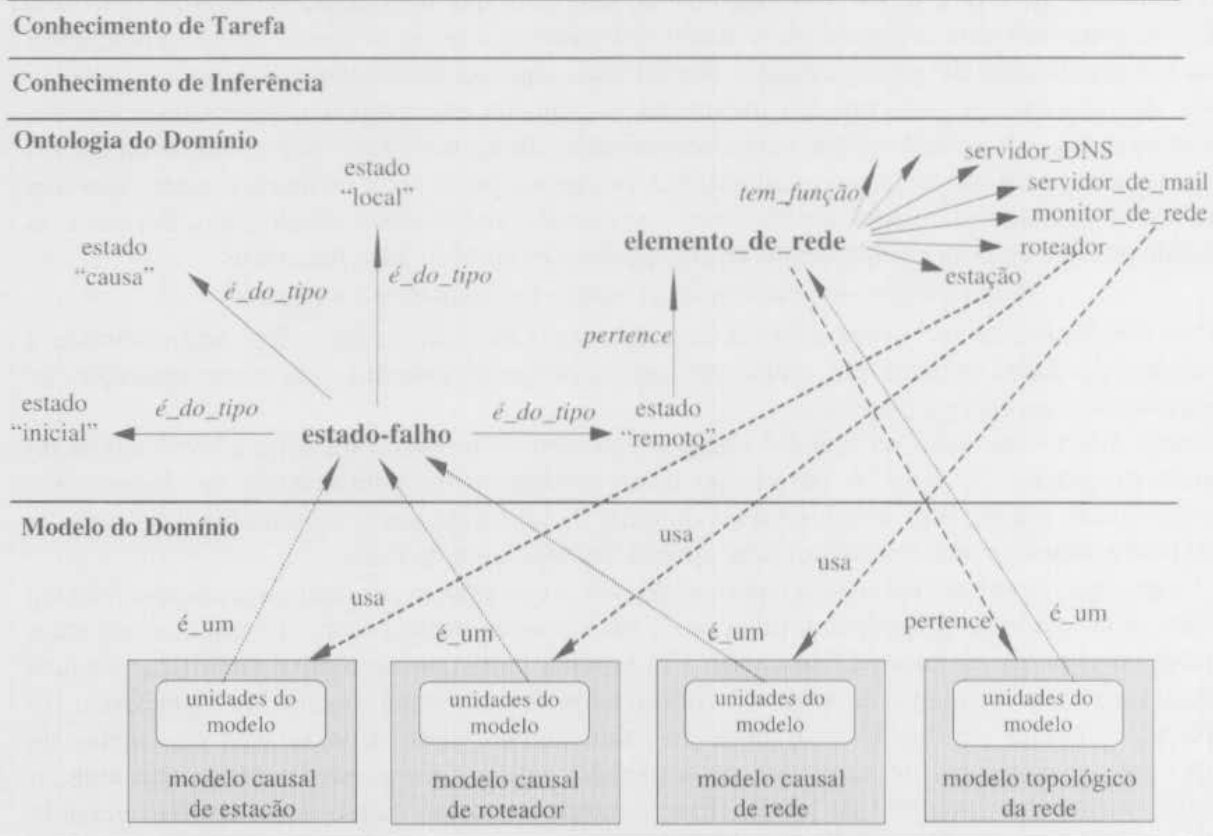


Figura 2 - Ontologia de domínio proposta para o domínio de gerenciamento de redes

Portanto, estados-falhos de um modelo causal podem ser classificados em quatro tipos:

- ❑ **estados-falhos inicial:** estados que são observados pelo método para detectar sintomas;
- ❑ **estados-falhos local:** estados que podem existir no elemento de rede observado [15];
- ❑ **estados-falhos remoto:** estados que podem existir em elementos remotos ao elemento de rede observado [15];
- ❑ **estados-falhos causa:** estados que, quando encontrados pelo método numa sessão de diagnóstico, representam o diagnóstico.

A figura 3, abaixo, mostra um modelo causal rotulado com conhecimento explicitado na ontologia de domínio.

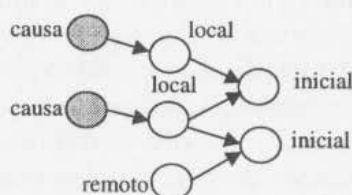


Figura 3 - Modelo causal com quatro tipos de estados-falhos: inicial, local, remoto e causa

Nem sempre será possível para o método encontrar um estado-falho, do tipo causa, através da análise de um único elemento de rede, ou seja, de um único modelo causal de estados-falhos. Pelo fato de que os elementos de rede interagem entre si, existirão situações em que o método deve analisar mais do que um modelo para encontrar o diagnóstico. Para isso, é necessária a existência de duas informações para que o método continue a busca pelo diagnóstico: (i) quais os elementos remotos disponíveis para análise; (ii) qual o estado-falho que deve ser inicialmente detectado nos elementos remotos. A primeira condição é satisfeita com a informação topológica da rede. A segunda condição é satisfeita modelando-se a interação entre estados-falhos de elementos de rede diferentes. Com essas duas informações, o método conhece exatamente qual o elemento de rede onde deve coletar eventos a fim de prosseguir a busca pelo diagnóstico.

Uma outra *unidade de modelo* identificada na ontologia de domínio é o *elemento de rede*, o qual é a unidade representativa do modelo topológico da rede e está caracterizada na ontologia de domínio através de sua função na rede. Essa caracterização funcional está relacionada com os modelos de estados, pois estes podem ser construídos segundo o aspecto funcional gerenciado. Por exemplo: roteamento, tráfego, serviço de *mail*, serviço DNS e outros. A tabela 1 relaciona alguns tipos de modelos que podem ser construídos e as MIBs que fornecem variáveis relevantes ao aspecto gerenciado, o qual determina uma funcionalidade para elementos de rede.

FUNÇÃO DO ER	TIPO DE MODELO	MIB (REDE TCP/IP)
<i>monitor-de-rede</i>	modelo de estados-falhos de rede	RFC 1757 (RMON)
<i>roteador</i>	modelo de estados-falhos de roteador	RFC 1213 (MIB-II)
<i>estação</i>	modelo de estados-falhos de estação	RFC 1213 (MIB-II)
<i>servidor de mail</i>	modelo de estados-falhos de serviços de Mail	RFC 1566
<i>servidor DNS</i>	modelo de estados-falhos de serviços DNS	RFC 1611

Tabela 1 - Possíveis tipos de modelos para gerenciamento de redes

A função *estação* é definida como uma função genérica, na qual uma estação é gerenciada sem considerar que tipos de aplicações rodam sobre ela. O modelo causal para essa função deve relacionar objetos/atributos gerenciados comuns a todas as estações que utilizam uma mesma tecnologia de rede. Para a função *roteador*, o modelo causal deve relacionar objetos/atributos capazes de gerenciar a funcionalidade de roteamento de pacotes. Para a função *monitor_de_rede*, o modelo causal deve relacionar objetos gerenciados/atributos capazes de fornecer informações gerais sobre a rede. Assim, uma biblioteca de modelos pode ser construída e introduzida num sistema de gerenciamento de tal forma que uma aplicação de diagnóstico possa selecionar modelos, em tempo de execução, de acordo com a função que elementos de rede exercem na topologia gerenciada. Cada função exige um conjunto de objetos gerenciáveis e atributos, capaz de representar estados-falhos importantes para a gerência da função e consequentemente do elemento de rede que exerce tal função.

Essa maneira de modelar o domínio permite a reusabilidade dos modelos em novas aplicações de diagnóstico e facilidade na alteração de um SBC, especificado e implementado desse modo. Se a topologia da rede é alterada, com a inserção ou retirada de elementos de rede,

basta disponibilizar para o sistema, o novo modelo topológico e informar a função dos novos elementos de rede.

Este trabalho não define uma ontologia exaustiva no domínio de redes. A idéia é, antes de tudo, mostrar a importância da identificação dos vários tipos de conhecimento e como eles podem ser modelados e reusados. A identificação e construção dos respectivos modelos causais de estados-falhos podem e devem ser realimentados por toda a comunidade do domínio.

A figura 4 mostra uma seqüência de estados encontrada pelo método, numa sessão de diagnóstico, envolvendo dois elementos de rede representados por seus respectivos modelos causais. A seqüência representa os efeitos provocados pela ocorrência do estado e9 no elemento de rede A. A busca pelo estado e9 (a causa-raiz) iniciou-se quando o estado e3 foi detectado no elemento de rede-B.

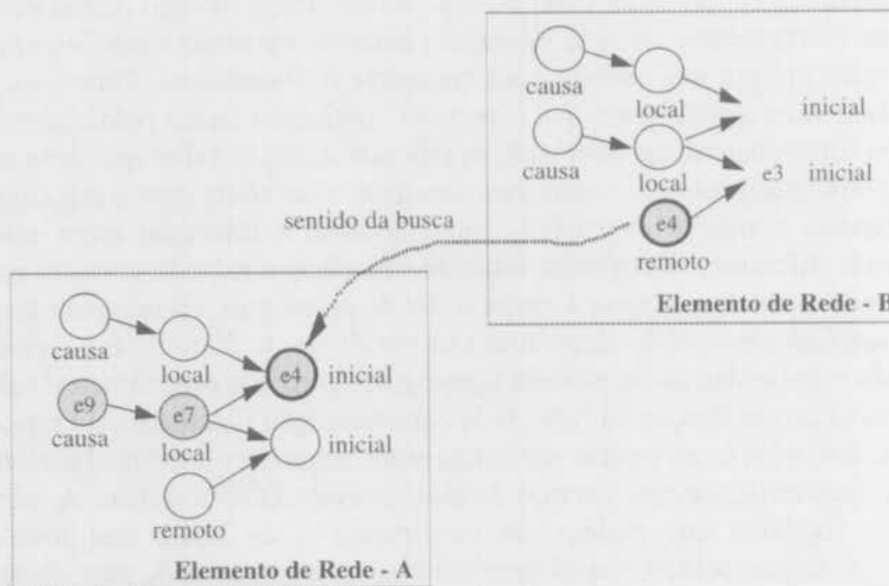


Figura 4 - Seqüência de estados obtida numa sessão de diagnóstico: e3, e4, e7 e e9.

Note que a ontologia de domínio não é uma estrutura pronta para ser utilizada no sistema, e sim um conhecimento a ser introduzido na especificação do método e na construção dos modelos causais. A próxima seção apresenta o PSM obtido, representado através de estruturas CommonKADS.

4 O MODELO DA TAREFA DIAGNÓSTICO PARA O DOMÍNIO DE GERENCIAMENTO DE REDES

Para efetivamente construir o PSM para o domínio de gerenciamento de redes, os métodos da biblioteca de Benjamins foram estudados, observando-se principalmente os critérios de adequabilidade dos métodos. Aqueles que se adequaram ao domínio foram selecionados e adaptados para compor o PSM proposto, trabalho este suportado pelas suposições descritas na seção 3.1 e no conhecimento ontológico explicitado na seção 3.2. O detalhamento dessa seleção, adaptação e funcionalidade de cada sub-tarefa encontram-se descritos em [11]. O PSM obtido está representado na estrutura de tarefas e métodos CommonKADS, na figura 5. No nível mais alto da decomposição da tarefa, o método *diagnóstico primário* decompõe a tarefa diagnóstico em três sub-tarefas: *detectar sintoma*, *gerar hipótese* e *discriminar hipótese*. Estas por sua vez, são decompostas recursivamente pelos métodos associados.

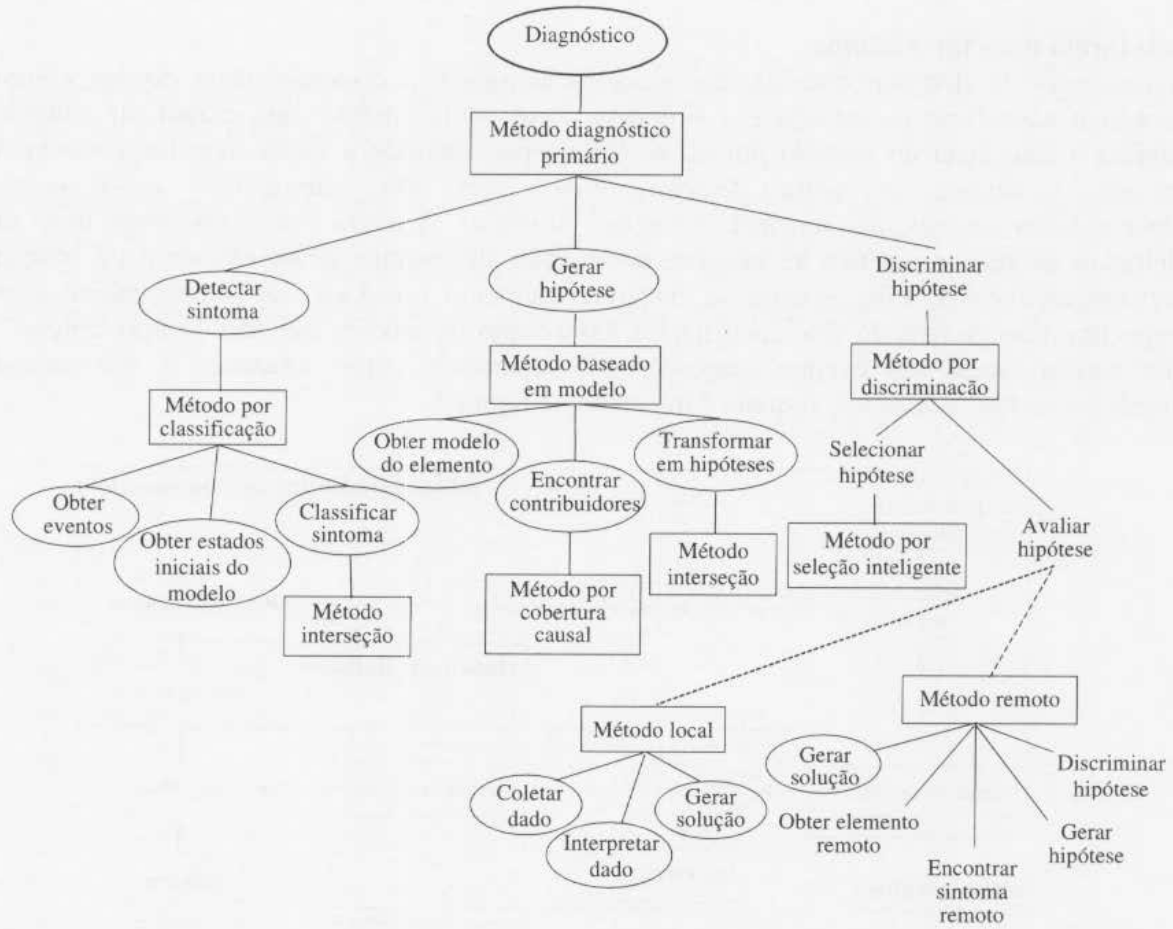


Figura 5 - Estrutura de decomposição de tarefas e métodos obtida para o domínio de gerenciamento de redes. Tarefas são representadas por elipses e métodos por retângulos. Linhas pontilhadas indicam que métodos são alternativos para alcançar o objetivo da tarefa. Linhas cheias decompõem um método em suas sub-tarefas.

A estrutura de tarefas e métodos não contém conhecimento de controle, apenas descreve como uma tarefa é decomposta por um método. Outra estrutura CommonKADS, a *estrutura de inferência*, descreve parte do controle que envolve as sub-tarefas na resolução do problema, identificando o fluxo de conhecimento na resolução da tarefa, através das conexões entre inferências e papéis do conhecimento, sem manipular diretamente os dados reais do domínio. A figura 6 apresenta a estrutura de inferência da tarefa de diagnóstico.

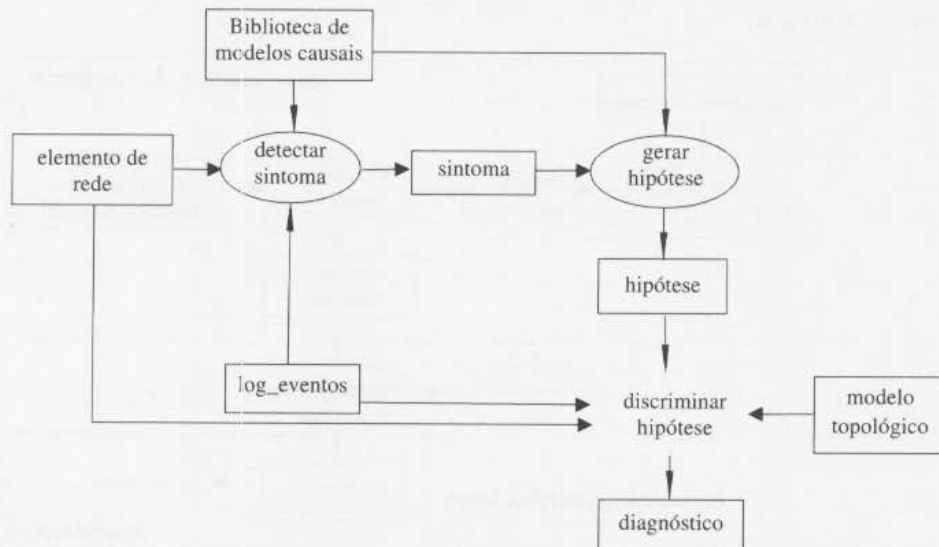


Figura 6 - Estrutura de inferência da tarefa diagnóstico. Retângulos correspondem aos papéis de conhecimento e elipses correspondem às inferências primitivas.

Sub-tarefa detectar_sintoma

A ontologia do domínio inserida nos modelos causais e a disponibilidade do *log_eventos* permitem classificar os estados em sintomas. Esse conhecimento para classificar sintomas habilita a utilização do método *por classificação* para realizar a tarefa *detectar_sintoma*. O método decompõe a tarefa *detectar_sintoma* em três sub-tarefas: *obter_eventos*, *obter_estados_iniciais_do_modelo* e *classificar_sintomas*. A tarefa *detectar_sintoma* toma um elemento de rede e verifica se existem ocorrências de eventos desse elemento na base de registros de eventos, o *log_eventos*. A função do elemento, que é alvo de gerenciamento, e seu respectivo tipo de modelo são identificados. Os estados do modelo que são do tipo “inicial” e que correspondem aos eventos ocorridos são assumidos como sintomas. A estrutura de inferência da tarefa *detectar_sintoma* é mostrada na figura 7.

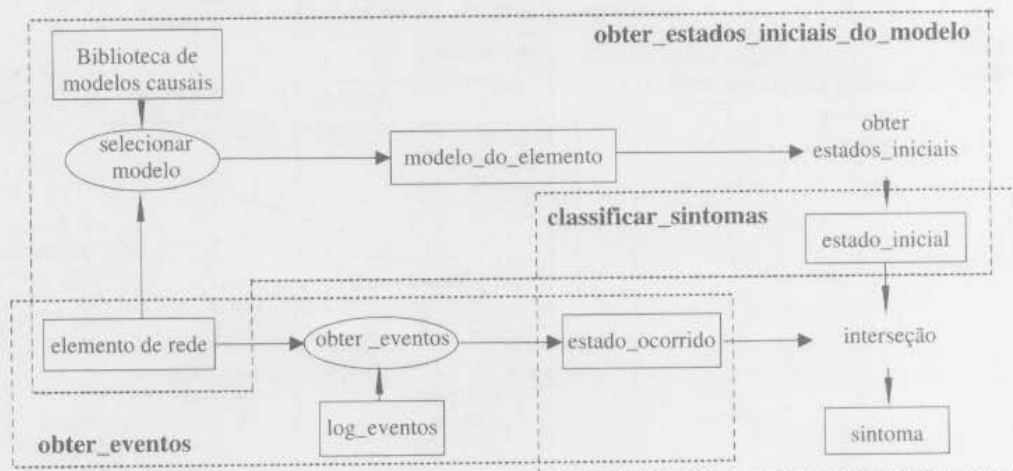


Figura 7 - Estrutura de inferência da tarefa *detectar_sintoma*

Sub-tarefa gerar_hipótese

Para realizar a tarefa *gerar_hipótese*, o método baseado em modelo é selecionado, uma vez que a representação da rede é composta de dois tipos de modelos: modelo causal e modelo topológico. O método baseado em modelo decompõe a tarefa *gerar_hipótese* em três sub-tarefas: *obter_modelo_do_elemento*, *encontrar_contribuidores* e *transformar_em_hipótese*. Na execução da tarefa *gerar_hipótese*, hipóteses são geradas percorrendo o modelo de estados do elemento de rede sob investigação, a partir dos sintomas. Hipóteses correspondem a estados do tipo “local”, “remoto” ou “causa”. A estrutura de inferência da sub-tarefa *gerar_hipótese* é mostrada na figura 8.

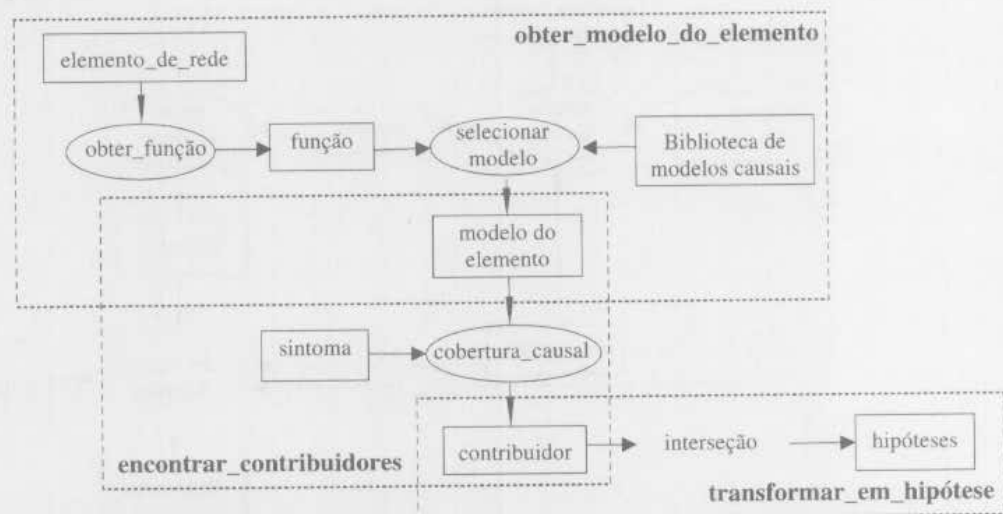


Figura 8 - Estrutura de inferência da tarefa *gerar_hipótese*

Sub-tarefa discriminar_hipótese

No modelo causal, qualquer estado falho representado, deve ser visto como um conhecimento estático e hipotético. Para verificar se um estado do modelo ocorreu na rede, a base de registros de eventos deve ser consultada. A sub-tarefa *discriminar_hipótese* tem esse objetivo. A biblioteca oferece uma decomposição que inclui três sub-tarefas: *selecionar_hipótese*, *coletar_dado* e *interpretar_dado*. Porém, no domínio de redes, sintomas podem ser observados num elemento de rede e a causa primária pode estar localizada em outro. Portanto, a decomposição de tarefas oferecida pela biblioteca foi adaptada para introduzir raciocínio sobre o modelo topológico da rede. A decomposição da tarefa *discriminar_hipótese*, consiste de duas sub-tarefas básicas: *selecionar_hipótese* e *avaliar_hipótese*.

A execução da tarefa *discriminar_hipótese* consiste da seleção de uma hipótese no conjunto de hipóteses realizada pela tarefa *selecionar_hipótese* e de sua validação pela tarefa *avaliar_hipótese*. Dois métodos podem ser necessários para a execução da tarefa *avaliar_hipótese*: método local e/ou método remoto. A introdução de conhecimento ontológico no sistema habilita a seleção dinâmica desses métodos para realizar a tarefa *avaliar_hipótese*. Se a hipótese é um estado "local", então ela pertence ao elemento de rede corrente, portanto o método local é selecionado. Se a hipótese é um estado "remoto", então ela pertence a outros elementos de rede, portanto o método remoto é selecionado. O método remoto consulta o modelo topológico da rede a fim de identificar o próximo elemento de rede a ser investigado. Os elementos de rede onde o estado remoto é detectado são analisados. Nesse estágio, o processamento volta a realizar a tarefa *gerar_hipótese* e novamente em *discriminar_hipótese*. A sessão de diagnóstico termina quando uma hipótese corresponde à um estado do tipo "causa" ou quando não existem mais elementos a serem investigados e a última hipótese resultou em "remota". Nesse último caso, a causa primária está localizada fora do domínio gerenciado. A rede causal que explica o estado "causa" encontrado, é formada pelos sintomas detectados no início da sessão de diagnóstico, pelas hipóteses válidas e as relações causais entre eles.

A figura 9 apresenta a estrutura de inferência da sub-tarefa *discriminar_hipótese* decomposta pelo método local e a figura 10 apresenta a estrutura de inferência da sub-tarefa *discriminar_hipótese* decomposta pelo método remoto.

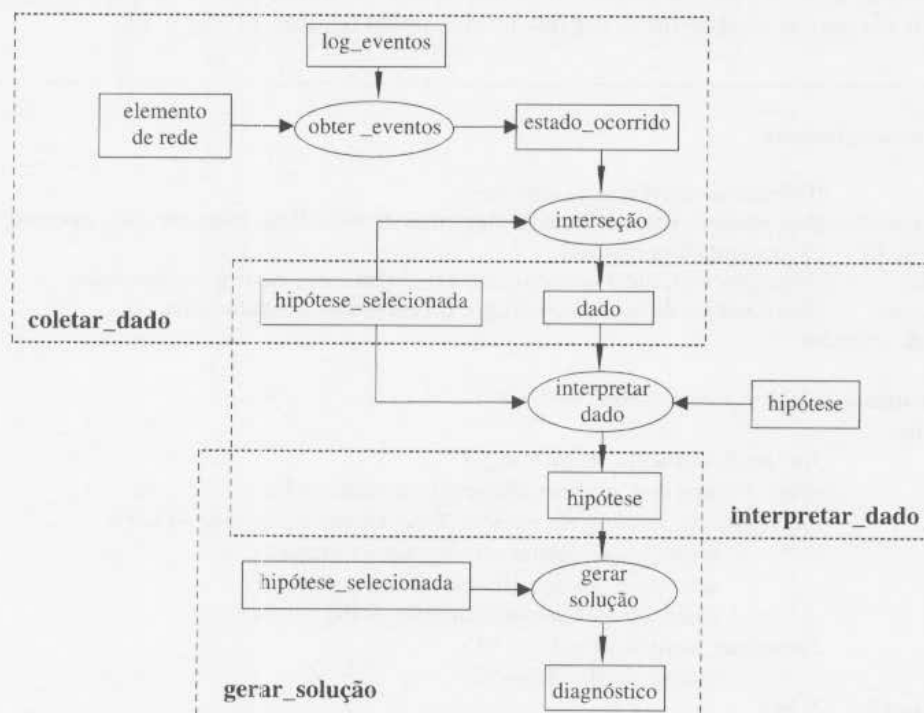


Figura 9 - Estrutura de inferência da tarefa *avaliar_hipótese* decomposta pelo método local

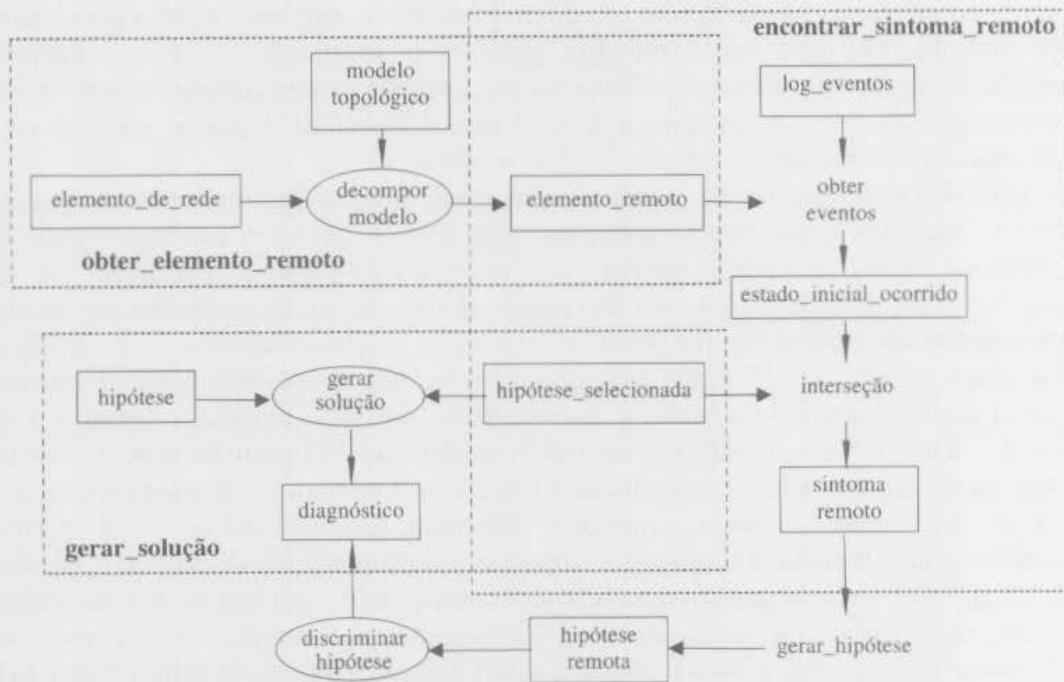


Figura 10 - Estrutura de inferência da tarefa *avaliar_hipótese* decomposta pelo método remoto

Após a construção das estruturas de inferência, adicionou-se controle sobre a execução das inferências e papéis de conhecimento incluindo *loops* e iterações. Assim obteve-se as *estruturas de controle da tarefa diagnóstica*, representadas pelas figuras 11, 12 e 13.

tarefa: *detectar_sintoma*

goal: [Detectar a ocorrência de sintomas]
 papel_de_entrada: [log_eventos, ER: conjunto de elementos de rede, Bmc: biblioteca_de_modelos_causais]
 papel_de_saida: [S: conjunto de sintomas]
 sub-tarefas: [obter_eventos, obter_estados_iniciais_do_modelo, classificar_sintomas]
 papel_interno: [Eo: conjunto de estados ocorridos, Ei: conjunto de estados iniciais]
 estrutura_de_controle:

detectar_sintoma (ER, log_eventos, Bmc → S) =

do

for each elemento_de_rede ∈ ER

obter_eventos (log_eventos, elemento_de_rede → Eo)

obter_estados_iniciais_do_modelo (Bmc, elemento_de_rede → Ei) =

obter_função (elemento_de_rede → função)

selecionar_modelo (função, Bmc → modelo)

obter_estados_iniciais (modelo → Ei)

classificar_sintomas(Eo, Ei → S) =

interseção (Eo, Ei → S)

until S != 0

Figura 11 - Estrutura de controle da tarefa *detectar_sintoma*

```

tarefa: gerar_hipótese
goal: [Gerar hipóteses que expliquem todos os sintomas]
papel_de_entrada: [S: conjunto de sintomas, Bmc: biblioteca_de_modelos_causais, elemento_de_rede]
papel_de_saída: [H: conjunto de hipóteses]
sub-tarefas: [obter_modelo_do_elemento, encontrar_contribuidores, transformar_em_hipótese]
papel_interno: [modelo_do_elemento, c: conjunto de contribuidores, C: conj. de conjuntos de contribuidores]
estrutura_de_controle:

gerar_hipótese (S, Bmc, elemento_de_rede → H) =
  obter_modelo_do_elemento (elemento_de_rede, Bmc → modelo_do_elemento)
  for each sintoma ∈ S
    encontrar_contribuidores (sintoma, modelo_do_elemento → c) =
      causal_covering (sintoma, modelo_do_elemento → c)
      C ← c
  end
  transformar_em_hipótese (C → H) =
    interseção (C → H)

```

Figura 12 - Estrutura de controle da tarefa gerar_hipótese

```

tarefa: discriminar_hipótese
goal: [Encontrar hipóteses que expliquem todos os sintomas e observações adicionais]
papel_de_entrada: [H: conjunto de hipóteses, elemento_de_rede, Mt: modelo_topológico, log_eventos]
papel_de_saída: [Diagnóstico/Diagnóstico_remoto: rede de estados que explicam os sintomas detectados]
sub-tarefas: [selecionar_hipótese, coletar_dado, interpretar_dado, gerar_solução, obter_elemento_remoto,
  encontrar_sintoma_remoto, gerar_hipótese, discriminar_hipótese]
papel_interno: [E: conjunto de estados ocorridos no elemento, ERR: conjunto de elementos remotos,
  hipótese_selecionada, sintoma_remoto, modelo_do_elemento, H_remoto, dado]
estrutura_de_controle:

discriminar_hipótese (H, elemento_de_rede, Mt, log_eventos → Diagnóstico) =
do
  selecionar_hipótese (H → hipótese_selecionada)
  if tipo (hipótese_selecionada, "local")
  then
    coletar_dado (elemento_de_rede, hipótese_selecionada, log_eventos → dado) =
      obter_eventos (elemento_de_rede, log_eventos → E)
      interseção (E, hipótese_selecionada → dado)
    interpretar_dado (dado, hipótese_selecionada, H → H)
  if tipo (hipótese_selecionada, "causa")
  then
    gerar_solução (hipótese_selecionada, H → Diagnóstico)
    retorna (Diagnóstico)
while ∃ hipótese para teste ∈ H ! tipo (hipótese_selecionada, "remoto")
if tipo (hipótese_selecionada, "remota")
then
  gerar_solução (hipótese_selecionada, H → Diagnóstico)
  obter_elemento_remoto (elemento_de_rede, Mt → ERR)
  for each elemento_remoto ∈ ERR
    encontrar_sintoma_remoto (elemento_remoto, log_eventos, hipótese_selecionada → sintoma_remoto)
    if (∃ sintoma_remoto)
    then
      gerar_hipótese (elemento_remoto, Bmc, sintoma_remoto → H_remoto)
      discriminar_hipótese (H_remoto, elemento_de_rede, Mt, log_eventos → Diagnóstico_remoto)
      Diagnóstico ← (Diagnóstico + Diagnóstico_remoto)
  until ER == 0
retorna (Diagnóstico)

```

Figura 13 - Estrutura de controle da tarefa discriminar_hipótese.

5 CONCLUSÕES

Este trabalho apresentou como técnicas do campo da Inteligência Artificial podem contribuir na solução de problemas da área de gerenciamento de redes de uma maneira cooperativa, reduzindo esforços e custo de projetos através da produção de modelos de conhecimento reusáveis. A abordagem de modelagem do conhecimento foi aplicada ao problema de gerenciamento de falhas por meio da metodologia CommonKADS que é uma metodologia bem estruturada e fundamentada nos conceitos dessa abordagem. Como resultado obteve-se a especificação de um método de resolução de problema (PSM) para a tarefa de diagnóstico de falhas em redes, descrito num alto nível de abstração - o nível do conhecimento - o que permite a sua reutilização na construção de SBCs que pretendam diagnosticar falhas para diferentes visões desse domínio. Obteve-se ainda uma proposta ontológica, representada pelos *papéis de conhecimento*, os quais definem o papel que elementos reais do domínio de gerenciamento de redes podem desempenhar na solução da tarefa. Esse é um importante resultado, uma vez que a quantidade de informações existentes no amplo e complexo domínio de redes dificulta a identificação da relevância das informações para a tarefa a ser realizada e a ontologia definida pode ser utilizada como um guia na aquisição de conhecimento do domínio durante a construção de aplicações de diagnóstico.

Uma característica importante a ser considerada em projetos de SBCs em gerenciamento de redes é a característica dinâmica das necessidades dos usuários. Elas refletem mudanças na topologia e nas aplicações que rodam na rede, o que pode implicar em mudanças nas aplicações de gerenciamento. A construção de SBCs dentro da abordagem de modelagem do conhecimento garante que mudanças nesses sistemas (os SBCs) sejam feitas de forma natural e flexível, pois cada tipo de conhecimento torna-se claramente explícito no modelo conceitual, facilitando a localização do conhecimento a ser alterado. Neste trabalho foi possível visualizar a real separação dos tipos de conhecimento envolvidos na construção de um SBC para gerenciamento de redes. Como o método manipula elementos do domínio através dos *papéis de conhecimento*, alterações na rede não refletem necessariamente mudanças no método embutido num SBC. Caso um novo elemento de rede seja inserido na rede, basta inseri-lo no modelo topológico e disponibilizar seu modelo causal para acesso pelo método, sem implicar em alterações no método ou modelos causais de outros elementos de rede.

Observou-se ainda, durante o desenvolvimento desta pesquisa, através de entrevistas realizadas com especialistas da área, a dificuldade em se identificar métodos de resolução de problemas utilizados por eles. Tais métodos nem sempre são bem compreendidos mesmo pelos próprios especialistas ou simplesmente ainda não existem métodos estabelecidos. Esse fato impossibilita a estruturação do conhecimento (de controle e do domínio) de outra maneira que não seja sob a abordagem baseada em regras de produção. Nesse sentido, o suporte da biblioteca de Benjamins foi de grande importância para a configuração do método proposto. Ela reúne esforço de anos de pesquisa no desenvolvimento de PSMs para diagnóstico, descritos de uma maneira reusável. Os critérios de adequabilidade da biblioteca direcionam o engenheiro do conhecimento na análise dos aspectos relevantes do domínio e que precisam ser considerados para a seleção dos métodos na biblioteca.

Atualmente, o método está sendo implementado no projeto GSAD para um estudo de caso especificado em [11]. A descrição no nível do conhecimento abstrai alguns detalhes do domínio, os quais serão verificados nesse trabalho em andamento. Isso permitirá futuros refinamentos e adaptações do método sobre o domínio. O desenvolvimento da pesquisa relatada é um primeiro passo em direção à construção e reuso de modelos de raciocínio e modelos de domínio para automação de tarefas complexas de gerenciamento de redes. O esforço conjunto da comunidade de gerenciamento de redes pode resultar, no futuro, numa biblioteca de modelos reusáveis específica para o uso em projetos de aplicações inteligentes para o domínio.

AGRADECIMENTOS

A pesquisa relatada aqui foi desenvolvida dentro do projeto "Gerenciamento de Sistemas de Alto Desempenho" (GSAD), da parceria LSI-EPUSP e ABC Bull S.A. Telematic. Marilza A. Lemos foi suportada por bolsa concedida pelo CNPq.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Abu-Hanna, A. *Multiple Domain Models in Diagnostic Reasoning*. Amsterdam, 1994. 169p. Thesis (PhD), University of Amsterdam.
- [2] Benjamins, V. R. *Problem Solving Methods for Diagnosis*. Amsterdam, 1993. 172p. Thesis (PhD), University of Amsterdam.
- [3] Bredeweg, Bert. *Expertise in Qualitative Prediction of Behaviour*. Amsterdam, 1992. Thesis (PhD), University of Amsterdam.
- [4] Breuker, J.; van de Velde, W. *CommonKADS Library for Expertise Modelling Reusable problem solving components*. IOS Press, Amsterdam, 1994.
- [5] Breuker, J. Components of Problem Solving and Types of Problems. In: EKAW - European Knowledge Acquisition Workshop, 8th, Belgium, Sep 1994, *Proceedings*, p. 118-136. (Lecture Notes in Artificial Intelligence).
- [6] Carvalho, T. C. M. B. *Gerenciamento de Redes: Uma abordagem de Sistemas Abertos*. BRISA, Makron Books do Brasil, 1993.
- [7] Clancey, W. J. Notes on "Epistemology of a rule-based expert system". *Artificial Intelligence*, v. 59, n. 1-2, p. 197-204, Feb, 1993.
- [8] Davis, R.; Buchanan, B. G.; Shortliffe, E. H. Retrospective on "Production rules as a representation for a knowledge-based consultation program", *Artificial Intelligence*, v. 59, n. 1-2, p. 181-189, Feb 1993.
- [9] de Barros, L. N.; Hendler, J.; Benjamins, V. R. Par-KAP: a Knowledge Acquisition Tool for Building Practical Planning Systems. In: IJCAI, 1997. *Proceedings*, 1997.
- [10] Fensel, D.; Straatman, R. Problem Solving Methods: Making Assumptions for Efficiency Reasons. In: EKAW'96 - European Knowledge Acquisition Workshop, 9th, Nottingham, United Kingdom, May, 1996. *Proceedings*, p. 17-32. (Lecture Notes in Artificial Intelligence, v. 1076).
- [11] Lemos, M.A. *Engenharia do Conhecimento Aplicada ao Domínio de Gerenciamento de Falhas em Redes de Comunicação: uma abordagem baseada em modelo*. São Paulo, 1997, 136p. Dissertação (Mestrado) - DAIA - EPUSP - Universidade de São Paulo.
- [12] Newell, A. The knowledge level. *Artificial Intelligence*, v. 18, p. 87-127, 1982.
- [13] Nunes, C. M. *Um Discriminador Inteligente de Eventos de Rede para o ambiente CINEMA*. Porto Alegre, 1997, 143p. Dissertação (Mestrado) - CPGCC, UFRGS.
- [14] Pras, A. *Network Management Architectures*, Enschede, The Netherlands, 1995, series n. 95-02, 191p. Thesis (PhD). University of Twente.
- [15] Rose, M. T.; McCloghrie, K. *How to Manage Your Network Using SNMP: the networking management practicum*. Prentice Hall, New Jersey, 1995.
- [16] Schreiber, G.; Wielinga, B; Breuker, J. *KADS: A Principled Approach to Knowledge-Based System Development*. Knowledge-Based Systems Book Series, v. 11, Academic Press, London, 1993.
- [17] Valente, A. *Legal Knowledge Engineering: A modelling approach*. Amsterdam, 1995. Thesis (PhD), University of Amsterdam.
- [18] Yemini, Y. *Network Management: Problems, Technologies, Standards - The State of The Art*. In: SBRC - Simpósio Brasileiro de Redes de Computadores, 14^o, Fortaleza, 20-23 maio, 1996. *Anais*. Fortaleza, SBC, 1996.