Suporte para vistas em ambientes de gestão de rede

Carlos Palma[1], Luís Rodrigues[2]

[1]Rural Informática, Damaia, 2720 Amadora {cpalma@ruralinf.pt}
[2]Faculdade de Ciências, Universidade de Lisboa, 1749 Lisboa {ler@di.fc.ul.pt}

Palavras chave: Vistas, Gestão de Rede, CRC2000.

Resumo

Em sistemas de gestão de redes, uma vista representa uma perspectiva parcelar do universo de gestão, adequada a uma actividade ou cargo específico. O suporte para vistas assume uma especial relevância em sistemas de grande dimensão onde não é possível concentrar toda a actividade de gestão. Este artigo aborda o problema, propondo um modelo de vistas que é concretizado num conjunto de extensões ao sistema Scotty, uma conhecida plataforma de gestão de rede de domínio público.

I. Introdução

A grande maioria dos actuais sistemas de gestão de rede permite manter uma descrição dos componentes da rede e seus atributos, bem como da representação da rede. Em redes com elevado número de componentes é geralmente difícil visualizar e manipular um espaço plano de componentes. É pois útil agrupar os componentes da rede em *vistas* que reflectem diferentes domínios de gestão.

Neste artigo propomos um modelo de suporte à representação de redes de computadores destinada a ambientes de gestão de rede que distingue a representação dos componentes da rede, da descrição das múltiplas relações em que esses componentes podem estar envolvidos.

Este modelo caracteriza-se por descrever de uma forma unívoca os componentes da rede, permitindo a sua múltipla agregação em vistas. As vistas têm a capacidade de adequar os componentes ao contexto da vista, através da restrição de acesso às propriedades dos componentes e da disponibilização de um mecanismo flexível de propagação e tratamento de eventos. O modelo proposto é concretizado numa extensão *Tcl* passível de ser integrada no sistema *Scotty*[1], uma conhecida plataforma de gestão de redes de domínio público, aumentando a sua capacidade de representação da rede.

Este artigo está organizado do seguinte modo: Na secção II são apresentadas as principais motivações que levaram à apresentação deste modelo. A secção III oferece uma descrição do modelo proposto, cuja concretização em Tcl é apresentada na secção IV. A avaliação do modelo é apresentada na secção V. A secção VI descreve as principais propostas de

representação de redes de computadores e faz a sua comparação com o modelo que propomos. As conclusões são apresentadas na secção VI.

II. Motivação

Uma rede informática complexa possui um elevado número de componentes (computadores, dispositivos periféricos, aplicações, etc.), cada qual caracterizado por um conjunto variado de atributos que são acedidos e manipulados nas actividades de gestão da rede. Apesar da informação que caracteriza uma rede complexa poder atingir dimensões consideráveis, raramente as actividades de gestão da rede informática, necessitam de toda a informação disponível. Pelo contrário, torna-se útil definir pontos de vista, ou simplesmente vistas, que seleccionam os componentes (e os atributos) relevantes para cada actividade específica de gestão da rede ou do sistema de informação.

Por exemplo, considere-se uma rede que possui máquinas executando o sistema operativo Unix e outras executando Windows NT, distribuídas fisicamente por diversas salas e ligadas a diferentes segmentos de rede IP. Para certas actividades, é interessante organizar uma vista contendo a informação de gestão de acordo com a distribuição física das máquinas, indicando quais os seus nomes e sistema operativo. Para outras, fará mais sentido organizar a informação por rede, seleccionando apenas um tipo de sistema operativo. Estas duas representações espelham a existência das múltiplas relações entre os componentes da rede. É importante que o sistema de gestão tenha isto em conta, permitindo por um lado, representar os componentes da rede, as suas propriedades e comportamento individual e por outro lado, apresentar as diferentes relações em que estes componentes se podem ver envolvidos e qual respectivo comportamento que cada componente adquire.

A existência de um mecanismo de vistas simplifica esta necessidade, uma vez que a vista pode ser utilizada para agregar e tornar visíveis as representações dos componentes da rede que se encontram envolvidos numa relação especifica (por exemplo: pertencem a um determinado segmento de rede ou são impressoras), filtrando as propriedades e comportamento dos componentes que é relevante ao contexto da vista.

Desta forma simplifica-se a interacção do utilizador com o sistema de gestão, uma vez que perante uma determinada actividade de gestão (gestão de um segmento de rede, gestão de sistemas Unix, etc.), só é apresentando ao seu administrador a informação relevante a essa actividade (componentes da rede, respectivas propriedades e comportamento, eventos relevantes, etc.). Além disso, disponibiliza-se um importante mecanismo de controlo de acesso à informação de gestão, melhorando a segurança do sistema de gestão de rede e consequentemente a segurança da própria rede de computadores.

Apesar de existirem múltiplas propostas que endereçam a representação da rede no âmbito da sua gestão, não encontramos, tanto quanto é do nosso conhecimento, nenhum modelo que cubra as funcionalidades anteriormente descritas. Foi essa a principal motivação para a constituição do modelo que seguidamente propomos.

III. Proposta de modelo de representação de redes de computadores

Esta secção propõe um modelo para concretizar a noção de vistas. O modelo pretende ser suficientemente versátil para permitir a representação de realidades de gestão complexas e, ao mesmo tempo, suportar a observação coerente dos objectos geridos mesmo quando se executam concorrentemente actividades de gestão sobre vistas distintas.

III.1 Entidades elementares

As entidades elementares, necessárias à representação da rede e seus respectivos componentes são os *dispositivos, elos, embaixadores, pastas* e *repositórios.*

Um ***Dispositivo*** representa um determinado objecto que se pretende gerir. Representa tipicamente um componente da rede (físico ou lógico), como uma máquina ou um processo. Cada elemento alvo da gestão é representado por um único dispositivo que incorpora os seus atributos e comportamento. O dispositivo é também conhecido no nosso modelo por elemento nuclear (*core-element*).

Um ***Embaixador*** representa um dispositivo numa determinada perspectiva do universo de gestão, reflectindo total ou parcialmente os atributos do seu dispositivo. Um *Dispositivo* pode ser representado por múltiplos *embaixadores* pertencentes ou não a múltiplas perspectivas do universo de gestão (vistas). O *Embaixador* permite a visualização e manipulação da informação (atributos) que caracteriza o seu respectivo dispositivo. Este componente é também conhecido por elemento.

Um ***Elo*** representa uma ligação física ou lógica entre dois ou mais dispositivos numa determinada perspectiva do universo de gestão. Um *elo* interliga *embaixadores* representando desta forma uma das múltiplas relações possíveis entre os dispositivos.

Uma ***Pasta*** agrega de *embaixadores* e de *pastas* que permitindo estruturar os objectos de gestão de acordo com um critério organizacional. Uma *pasta* pode assumir uma dualidade de papeis: entidade agregadora de entidades ou entidade representante de dispositivos complexos.

Uma ***Vista*** é conjunto coerente de pastas, embaixadores e elos que pretendem representar uma das múltiplas perspectivas da rede que se pretende gerir. Uma vista tem associadas restrições que permitem concretizar políticas de controlo no acesso aos atributos dos dispositivos, como forma de aproximar o comportamento dos seus embaixadores ao contexto da vista. Esta entidade pode ainda ter associados filtros que definem as propriedades que os dispositivos deverão ter para que os seus embaixadores possam estar associados às vistas.

Um ***Repositório*** é o componente do sistema que armazena os restantes elementos que vimos anteriormente. Um repositório contém todos os representantes dos elementos da rede

(dispositivos) e todas as representações ou perspectivas em que esses elementos podem estar envolvidos (embaixadores, elos, pastas e vistas).

As entidades elementares são caracterizadas através de atributos. Distingue-se neste modelo dois tipos de atributos: atributos simples e métodos.

Os ***Atributos simples*** são variáveis de tipo elementar que representam propriedades das entidades, como por exemplo inteiros ou cadeias de carácteres. Estes atributos podem ser utilizados para associar informação administrativa às entidades, para conter informação necessária ao acesso ao elemento da rede, para especificar parâmetros que controlam funções de fiscalização ou de teste do elemento, para armazenar informação provisória relativa aos processos de fiscalização, etc.

Os ***Métodos*** são atributos associados a programas/funções. O valor de um atributo corresponde ao resultado da execução do respectivo programa. Estes programas podem ser executados a pedido (quando é solicitado o valor do atributo) ou com uma periodicidade pré-definida. Estes métodos podem ser utilizados para a disponibilização de atributos complexos resultantes da computação de atributos simples ou de atributos externos à entidade. Podem ainda ser utilizados para alterar, de uma forma periódica ou a pedido, os atributos das entidades do repositório ou mesmo actuar sobre os elementos da rede.

As entidades elementares relacionam-se do seguinte modo:

- Um dispositivo possui um ou mais atributos (simples e métodos) e referências para todos os seus embaixadores. Esta entidade possui a informação necessária para propagar eventos gerados no contexto de uma vista, para as restantes vistas em que está representado.
- Um embaixador possui uma referência para o dispositivo que representa, uma referência para a pasta em que está inserido e referências para os elos a que está associado. Os embaixadores não dispõem de atributos próprios: reflectem os atributos do dispositivo a que se encontram associados.
- Um elo possui um ou mais atributos e referencias para os embaixadores que interliga.
- Uma pasta possui um ou mais atributos, a referência para a pasta em que está inserida (ou para a vista, caso se trate de uma pasta raiz) e uma lista de referências para as pastas, elos e embaixadores que contém.
- Finalmente, uma vista possui um ou mais atributos, uma ou mais pastas, embaixadores ou elos.

O modelo de objectos encontra-se ilustrado na Fig. 1.

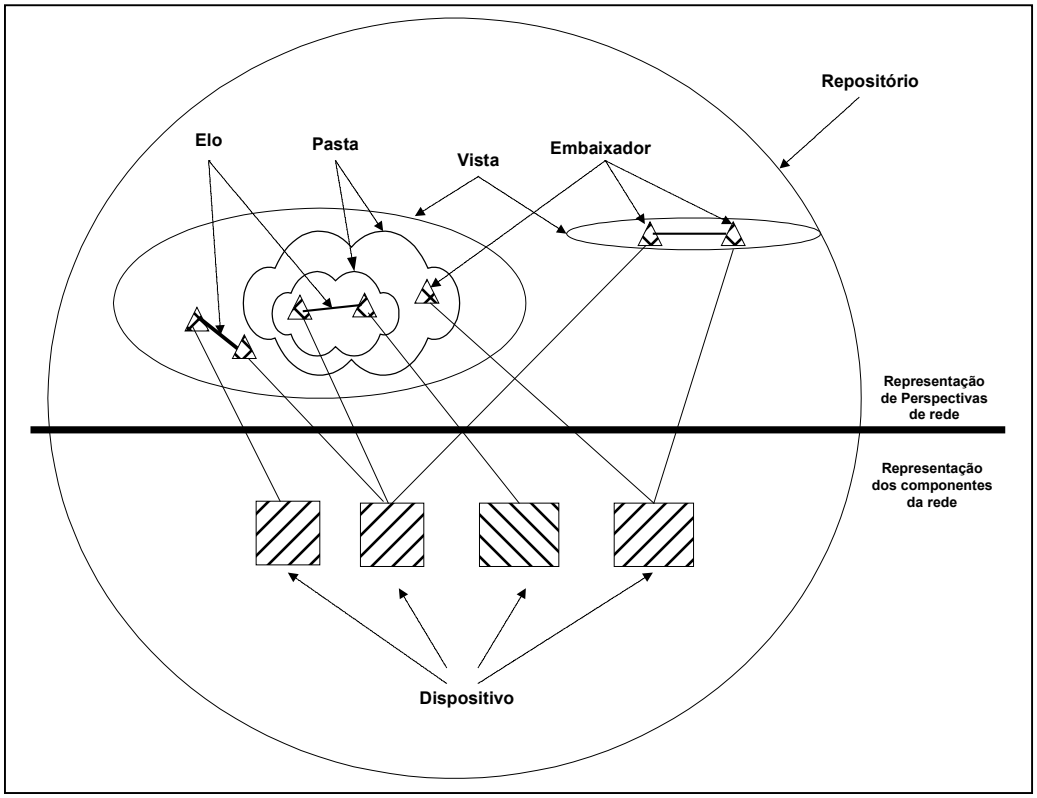

Fig. 1: Representação do modelo de entidades proposto

III.2 Restrições

Por defeito, um embaixador apresenta todos os atributos que caracterizam o dispositivo a que se encontra associado. O sistema que propomos, permite que sejam associados à vistas restrições, materializadas em listas de acesso, que limitam os atributos devem ser visíveis/manipuláveis nos componentes da vista (embaixadores, elos, pastas). Desta forma consegue-se adequar a caracterização dos componentes de uma vista ao seu respectivo contexto.

III.3 Filtros

Vimos anteriormente que um embaixador é um representante de um dispositivo numa vista. A criação/remoção de embaixadores numa vista pode ser efectuada de forma explicita ou implícita.

A criação explicita de embaixadores corresponde à criação dos embaixadores, no contexto de uma vista, associando-os aos respectivos dispositivos. Ou seja, um dispositivo só se encontra representado numa vista quando for expressamente determinado.

A criação implícita de embaixadores é efectuada através da definição na vista de um conjunto de regras (filtro) que estabelecem os valores que determinados atributos dos dispositivos devem conter para que estes possam ser representados na vista. Quando os atributos de um dispositivo cumprem a especificação do filtro, é criado na respectiva vista um embaixador. De igual modo, o embaixador é removido de uma vista quando os atributos do dispositivo que representam deixarem de cumprir as especificações do filtro.

III.4 Modelo de eventos

O comportamento das entidades é gerido pelo processamento de eventos. O nosso modelo possibilita a tipificação dos eventos e a respectiva associação de procedimentos ao nível de cada entidade (dispositivo, pasta, embaixador, etc.), permitindo a caracterização do comportamento de cada entidade perante a ocorrência de eventos. Os eventos podem tipificar-se em duas classes distintas: eventos internos e eventos externos.

Os eventos internos reflectem acontecimentos associados à manipulação da ferramenta, como seja, alteração de atributos, adição e remoção de entidades a uma vista, etc. O tratamento a este tipo de eventos só se pode verificar no contexto da entidade onde o evento ocorreu.

Os eventos externos reflectem acontecimentos associados ao universo alvo da gestão. Dividem-se em três subclasses: eventos ascendentes, eventos descendentes, eventos laterais. O tratamento dos eventos externos pode propagar-se a mais do que uma entidade, estando a forma de propagação necessariamente relacionada com a subclasse de eventos.

Caso ocorra um evento ascendente numa determinada entidade, o seu tratamento inicia-se nessa mesma entidade propagando-se pelas hierarquias superiores da representação (pastas que contêm directa ou indirectamente o elemento onde ocorreu o evento). No caso de ocorrência de um evento descendente, o seu tratamento inicia-se na entidade visada e propaga-se por todas as hierarquias inferiores, podendo terminar ao nível do dispositivo. Finalmente, a ocorrência de uma evento lateral o seu tratamento é propagado através da malha definida pelos elos a partir da entidade origem. Esta propagação é limitada por um número de saltos (*hops*) pré-definido.

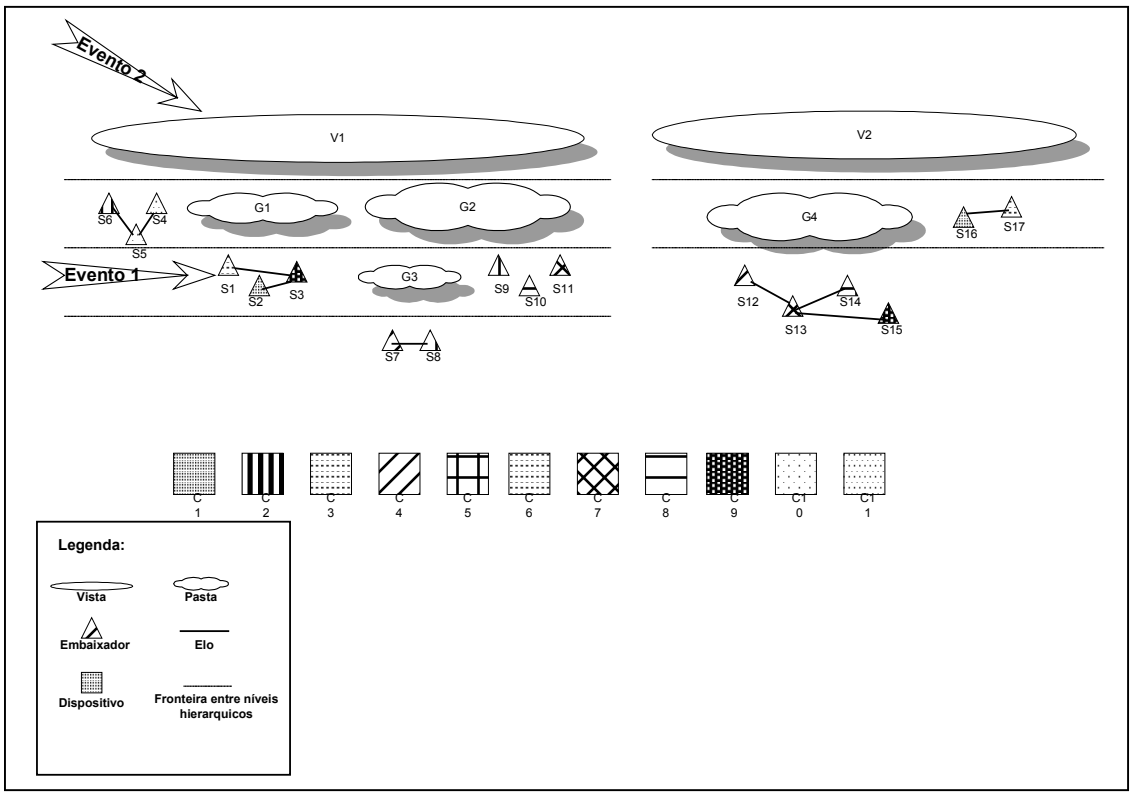

Fig. 2: Exemplo dos mecanismos de propagação de eventos

A Fig. 2 permite ilustrar o funcionamento deste mecanismo de eventos. A ocorrência do Evento 1 no embaixador S1 poderá gerar a três comportamentos distintos:

- Se o Evento for ascendente poderá ser tratado no embaixador S1, propagado e eventualmente tratado na pasta G1 e posteriormente propagado e eventualmente tratado na vista V1. O facto de o evento ser tratado ou não em cada uma das entidades por onde é propagado estará directamente relacionado com a necessidade de refinar o tratamento de determinada ocorrência. É assim possível, o tratamento genérico de uma determinado evento ao nível de uma vista ou de uma pasta ou um tratamento mais especializado ao nível do embaixador ou do dispositivo.
- Se o Evento 1 for descendente poderá ser tratado no embaixador S1 e posteriormente tratado no dispositivo C3. A ocorrência de eventos ao nível dos dispositivos tem a particularidade de gerar um evento ascendente que se propaga por todos os embaixadores do dispositivo. Caso o evento tenha origem num embaixador, este último não é alvo do evento ascendente gerado pelo seu dispositivo.

- Se considerarmos que o Evento 1 é lateral com um salto, será propagado para o embaixador S3.

A ocorrência do Evento 2 ao nível da vista V1 só fará sentido se estivermos na presença de um evento descendente. Neste caso, o evento será propagado e eventualmente tratado por todas as entidades pertencentes à vista V1. Este evento será ainda propagado para todos os dispositivos relacionados com os embaixadores da vista V1. Por sua vez, os dispositivos afectados pelo evento 2 darão origem a um evento ascendente que no caso concreto do exemplo apresentado, irão afectar as entidades da vista V2 (embaixadores, pastas e a vista V2). De notar que cada entidade que é "visitada" pela propagação de um evento tem a capacidade de, no seu tratamento, interromper a propagação.

Os mecanismos de propagação de eventos propostos permitem:

- Reflectir, de uma forma selectiva, o estado dos elementos da rede nas suas diferentes representações (Eventos ascendentes) .
- Tratar de agrupamentos de entidades sem que seja necessário um conhecimento exaustivo das mesmas (Eventos descendentes).
- Reflectir e tratar a alteração de estado dos elementos da rede que se encontram relacionados entre si (eventos laterais).

IV. Concretização em Tcl

De modo a validar o modelo proposto, foi desenvolvido um protótipo, materializado numa extensão Tcl, passível de ser integrada na plataforma Tnm/Scotty.

O sistema Scotty é um sistema de gestão de redes de domínio público que oferece uma interface para a linguagem Tcl[2][3]. A possibilidade de desenvolver aplicações de gestão personalizada usando uma linguagem interpretada como o Tcl, aliada ao facto de o Scotty se tratar de um sistema aberto, têm contribuído para a popularidade deste sistema. O Scotty dispõe de uma extensão Tcl, o *Tnm::map*[11], destinado à representação da rede que não dispõe de um mecanismo robusto de definição e gestão de vistas. O presente trabalho pretende aumentar o actual sistema Scotty com um sistema versátil de gestão de vistas baseado no modelo anteriormente apresentado.

Assim, optou-se nesta concretização pela expansão da extensão *Tnm::map*[11] em vez do desenvolvimento de raiz de uma nova extensão Tcl. Foi criada uma nova extensão Tcl, o *Tnm::repo*, que mantém as especificações da extensão original, tendo-se adicionado um conjunto de funcionalidades que permitem a concretização do modelo anteriormente descrito. Na definição da interface do *Tnm::repo* houve a preocupação de manter inalteradas as funcionalidades e interface presentes no *Tnm::map*, com o objectivo de permitir a coexistência de procedimentos desenvolvidos com recurso à duas extensões. A Figura 3 apresenta o modelo de dados da extensão *Tnm::repo*.

A terminologia do modelo e da interface do *Tnm::repo* apresentam as diferenças inerentes à adopção respectivamente da língua portuguesa e da língua inglesa. Assim, no *Tnm::repo*, o

repositório é designado por *repo* (*repository*), a vista por *map*, o dispositivo por *celem* (core-element), a pasta por *group*, o embaixador por *elem* (*element*), e o elo por *link*.

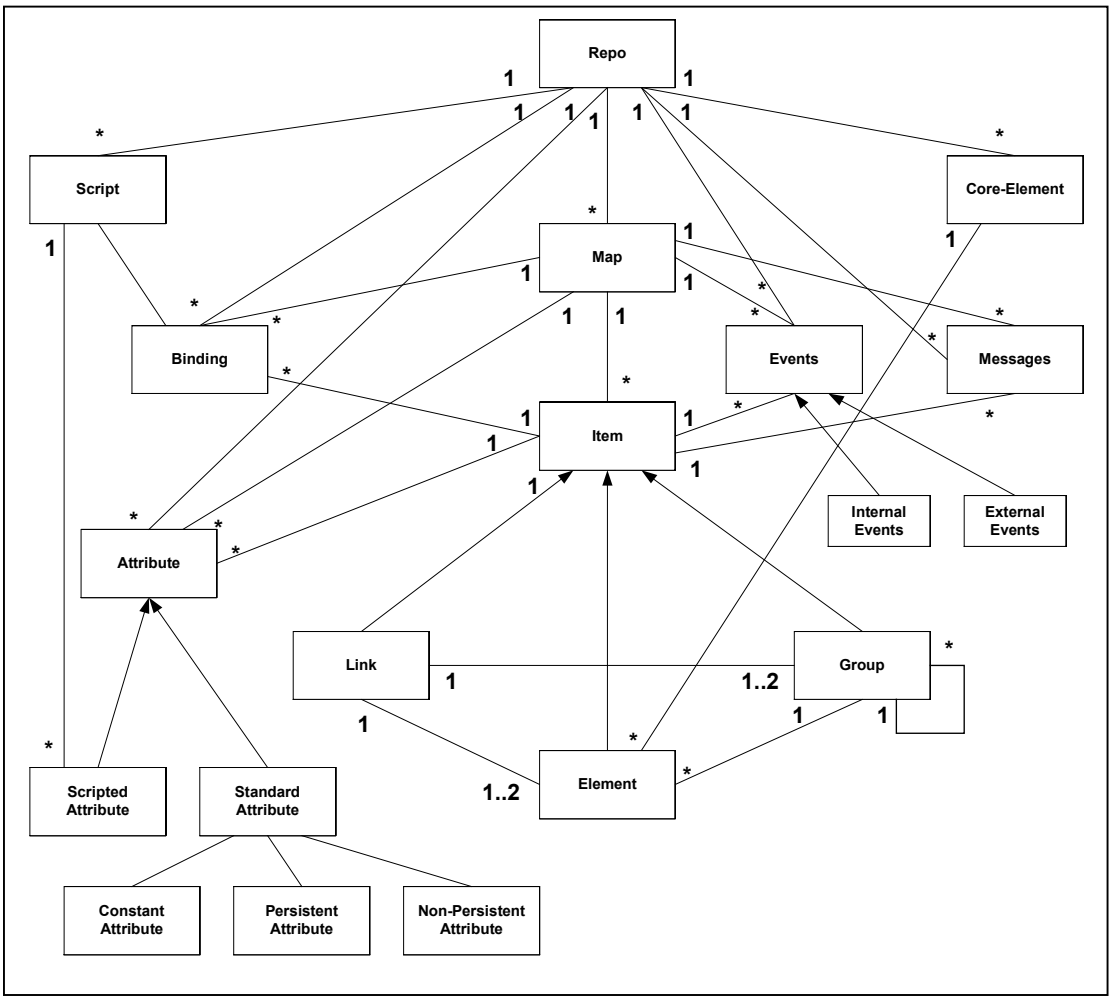

Fig. 3: Modelo de dados do Tnm::repo

A representação de uma rede é iniciada com a criação de um repositório (*repo*) que será responsável pela agregação e armazenamento de todas as entidades envolvidas. O repositório mantém o controlo directo sobre os dispositivos – *celem* , procedimentos – *scripts* (concretizam os métodos e o tratamento de eventos) e as vistas -*maps.* Sobre um repositório pode-se efectuar um conjunto de operações de manutenção da representação.

Vejamos os seguintes exemplos:

```
# Criação de um repositório
set rede1 [Tnm::repo create –name rede1]

# Criação de dois dispositivos (Um PC e uma impressora)
set pc1 [$rede1 create celem –name pc1 –subtype pc1]
set pr1 [$rede1 create celem –name pc1 –subtype printer1]

# Criação de uma vista explicita
set view1 [$rede1 create map –name vista1]

# Criação de uma vista implícita
set view2 [$rede1 create map –name vista2 –filter [list :SysType == pc]

# Criação de um procedimento
set scr1 [$rede1 create script –name function1 –script {
                    if { “%T” == “repo” } {
                            set function1 true
                    } else {
                            set function1 false
                    }
          }

# Tratamento de atributos simples
$rede1 attribute set :Global:Teste1 “teste 123”
set result [$rede1 attribute get :Global:Teste1]
$rede1 attribute unset :Global:Teste1

# Definição de métodos
$rede1 attribute set :Global:Var1 –script $scr1

# Recolha de informações relativa ao repositório
set vistas [$rede1 info maps]
set dispositivos [$rede info celems]

# Remoção do repositório
$rede1 destroy
```

Cada componente da rede (físico ou lógico) é representado univocamente por um dispositivo (*celem*). Esta entidade é responsável por manter toda a informação sobre o componente que representa. Sobre o dispositivo podem efectuar-se operações de manutenção de atributos, obtenção de informações (repositório associado, embaixadores associados, etc.) e mesmo a remoção da entidade como se poderá observar no exemplo seguinte:

```
# Tratamento de atributos simples
$pc1 attribute set :SysType == pc
set result [$pc1 attribute get :SysType]
$pc1 attribute unset :Global:Teste1
```

A definição dos procedimentos (*scripts*) utilizados nos métodos e no tratamento de eventos ao nível do repositório tem a vantagem de permitir a reutilização de código e simplificação das tarefas de manutenção. Estes procedimentos são interpretados em tempo de execução e podem conter chaves de substituição (sinal de % seguido de um carácter) que são substituídas por valores quando o procedimento é invocado. Estas chaves permitem que o procedimento tenha acesso a informações tais como: Testemunho da entidade associada ao procedimento (%H), tipo de entidade associada ao procedimento (%T), nome do atributo (%V), etc. No procedimento *function1* definido anteriormente, a chave %T é substituída em tempo de execução por "*repo*".

A representação de perspectivas distintas da rede é efectuada através respectiva criação de vistas. Sobre uma vista (*map*) suporta operações tais como a manipulação de atributos, criação de itens (embaixadores, pastas e elos), obtenção de informações, etc.:

```
# Criação de embaixadores
set item1 [$view1 create elem $pc1 –name pc01]
set item2 [$view1 create elem $pr1 –name pr01]

# Criação de um grupo
set gr1 [$view1 create group –name group1]

# Criação de um elo
set lnk1 [$view1 create link –name wire1]

# Obtenção de informações
set listaelems [view1 info elems]
set listaatributos [view1 info attributes]

# Remoção de uma vista
#view1 destroy
```

As vistas explicitas distinguem-se das vistas implícitas no facto de ser possível, nas primeiras, a criação de embaixadores. Tal como foi referido anteriormente, os embaixadores associados a vistas implícitas são definidos automaticamente com base nas regras definidas no filtro associado à vista.

A restrição no acesso a atributos dos dispositivos, ao nível dos seus embaixadores, é efectuada através da definição de restrições ao nível das vistas respectivas:

```
# Inibir a visualização nos itens de atributos iniciados por :Global:Secret
$view1 configure –constraints [list deny :Global:Secret]
```

Os itens servem de base à representação das relações entre os componentes da rede sob uma determinada perspectiva. Por exemplo, podem representar a interligação entre equipamentos de encaminhamento de pacotes, ou a agregação de equipamentos por domínios Windows NT, etc.

Sobre os itens podem efectuar um conjunto de operações que são independentes do tipo de item: manipulação de atributos, obtenção de informações (vista e repositório a que pertencem, etc.).

Existem ainda operações especializadas a determinados tipos de itens. Sobre os embaixadores é possível obter o testemunho dos respectivos dispositivos associados; os embaixadores e as pastas possibilitam a definição e uma pasta mãe caracterizando desta forma um hierarquia de entidades; os elos permitem a definição/obtenção das entidades a eles ligados. Vejamos alguns exemplos:

```
# Associação de um item a uma pasta
$item1 configure  -group $gr1

#  Obtenção dos membros de um grupo
set membros [$gr1  info members]

# Interligação de dois itens através de um elo
$lnk1 configure –src $item1 –dst  $item2

# Obtenção dos itens que se encontram associados a um elo
set item_origem [$lnk1 cget –src]
set item_destino  [$lnk1 cget –dst]
```

A extensão *Tnm::repo* suporta programação orientada a eventos. O tratamento de eventos, extensivo a todas as entidades elementares, é composto por duas fases distintas: definição do evento que se pretende tratar e notificação de ocorrência do evento.

Na definição do evento a tratar é efectuada uma associação (*binding*) entre o respectivo identificador e o procedimento (script), entretanto definido ao nível do repositório, que deverá ser executado sempre que o evento ocorra. A notificação de ocorrência de um evento é efectuada através da execução do comando "*raise"*. O exemplo seguinte apresenta o tratamento de um evento ascendente.

```
# Criação de um procedimento de tratamento de eventos
set scr2  [$rede1 create script –name trataevt1 –script {
                    if { %T == elem  } {
                            %H attribute set :Global:Erro True
        }
}

# Definição do tratamento do evento ascendente
$item1 create bind :Eventos:Erro1 –script $scr2 –up

# Notificação de ocorrência de um evento
$item1 raise –up :Eventos:Erro1
```

De notar que somente os eventos externos podem ser accionados através do comando "*raise*". Os eventos internos, reconhecidos através de um conjunto de identificadores pré-definidos, não podem ser accionados externamente à entidade, o que torna inválidos os seguintes comandos:

```
$rede1 raise -up :Repo:InternalEvents:CreateCelem
$item1 raise -down :Item:InternalEvents:SetAttribute
$view1 raise -up :View:InternalEvents:CreateElem
```

O modelo que propomos adequa-se à representação dos componentes de uma rede e suas respectivas interdependências, tendo em vista a gestão da rede. No entanto, caracteriza-se por ser independente dos protocolos de gestão de rede que permitem a obtenção de informação e actuação sobre os componentes da rede. A integração entre o *Tnm::repo* e os protocolos de gestão de rede é efectuada ao nível dos procedimentos (*scripts)* que podem utilizar as facilidades disponibilizadas por outras extensões TCL com especial destaque para o *Scotty* (SNMP, ICMP, HTTP, *sockets*, etc). Esta independência possibilita que o utilizador de uma vista se abstraia dos protocolos de gestão da rede focando-se na rede e no seu comportamento.

V. Avaliação

De modo a avaliar o modelo aqui proposto aplicámos o *Tmn::repo* a uma rede alvo e comparamos o resultado com a aplicação de outros modelos alternativos, especificamente do *Tmn::map* e do *HP OpenView*, à mesma rede. Fazemos aqui um breve sumário dos resultados desta comparação, a qual se encontra descrita em [12].

A rede alvo foi escolhida de modo a incluir diferentes componentes interligados que podem ser observados por três vistas diferentes: a vista dos gestores de plataformas *MS Windows*, a vista dos gestores de plataformas *Unix* e a vista dos gestores dos equipamentos de interligação. Alguns componentes, como por exemplo os encaminhadores por omissão, necessitam de ser representados em mais que uma vista, apesar da informação a apresentar aos seus respectivos gestores seja distinta.

Este exercício torna patente algumas vantagens do modelo aqui proposto. Por exemplo, no *HP OpenView* não existe nenhum modo de restringir a visualização e manipulação de atributos de um dado objecto nas diferentes vistas. Para obter esta funcionalidade no *Tmn:map*, foi necessária criação de vários mapas cuja coerência só é assegurada se o utilizador programar explicitamente um conjunto de procedimentos para relacionar a informação contida nessas estruturas. Pelo contrário, a clara separação entre vistas e os dispositivos do nosso modelo veio permitir representar as três vistas de gestão sem nenhumas destas limitações: cada gestor obtém somente a informação de que necessita.

VI. Trabalho Relacionado

A necessidade de agrupamento dos objectos alvo da gestão tem acompanhado a evolução da gestão dos sistemas. A disponibilização de mecanismos de suporte à representação da rede encontra-se presente na maior parte dos sistemas de gestão, embora com objectivos e abordagens distintas.

Nos sistemas de gestão de rede vigentes podem tipificar-se duas classes de soluções de representação da rede: soluções cuja unidade básica de representação é o componente da rede e soluções que têm como base os atributos desses mesmos componentes.

A primeira classe de soluções tem como objecto de tratamento os elementos da rede que se caracterizam por um conjunto de atributos. Nesta classificação, podemos englobar desde

soluções do domínio público a sistemas de ``gestão empresarial''. O Tkined [4] e o Scotty, são bons exemplo de soluções de domínio público. O *CA Unicenter* [5] e o *HP OpenView* [6] são exemplos de soluções comerciais cuja gestão assenta no elemento da rede. Engloba-se ainda nesta classe de soluções o sistema *Domains* [7].

A segunda classe de soluções refina a representação ao considerar os atributos dos elementos como unidade básica. Neste tipo de soluções, cada componente da rede é visto como o agrupamento de um determinado conjunto de atributos. É possível a criação de representações resultantes do agrupamento de atributos que não correspondem a nenhum dos elementos reais da rede. Se na primeira classe de soluções e no que diz respeito à semântica da representação da rede não se verifica grandes diferenças de conceito entre elas, o mesmo não se passa nesta segunda classe. Verificam-se aqui grandes diferenças de representação sendo possível encontrar desde representações de rede baseadas em folhas de cálculo [8] a soluções que permitem interagir com a representação da rede através de SQL (*SNMPQL* [9]) , passando por soluções cujas vistas são agrupamentos de atributos (*Marvel* [10]).

A nossa concretização, o *Tnm::repo,* engloba-se claramente na primeira classe de soluções. Deste modo, centramos o nosso estudo comparativo em dois sistemas comerciais representativos. Consideremos os sistemas o *CA Unicenter TNG* e o *HP OpenView* que caracterizam a primeira classe de soluções e que mostram qual a abordagem típica de representação da rede.

O *CA Unicenter* representa a rede através um modelo orientado a objectos. Os componentes básicos de representação da rede são as classes e os objectos. As classes definem os diferentes tipos de elementos da rede, suas propriedades (atributos e relações) e comportamentos. Os objectos, instancias de uma determinada classe, representam um elemento concreto da rede. No *Unicenter TNG*, a representação da rede é suportada por dois modelos: modelo de objectos e modelo de topologia. No modelo de objectos são definidas as classes e respectivas instâncias. Este modelo preocupa-se com a caracterização dos atributos e comportamento dos elementos da rede. O modelo topológico tem como função definir as relações entre objectos. O facto de ser separado do modelo de objectos permite que um objecto possa estar representado em múltiplas relações com outros objectos.

No sistema *HP OpenView* verifica-se uma distinção entre a componente apresentação e a componente representação dos objectos. Estes últimos, designados nesta plataforma simplesmente por objectos encontram-se agregados em repositórios denominados mapas. Os objectos não são directamente visíveis, sendo representados na componente de apresentação por símbolos. São estes elementos que contêm a informação de apresentação de um objecto. O objecto pode ser representado por um ou mais símbolos. Os símbolos são agregados em sub-mapas e podem estar organizados de uma forma hierárquica (um mapa contém no mínimo um sub-mapa). Nesta plataforma o conceito de vistas encontra-se materializado nos sub-mapas servindo os símbolos de elo de ligação com os objectos.

Em qualquer destes sistemas de gestão, a representação dos objectos não é sensível à vista, ou seja, os atributos e as operações passíveis de serem efectuadas sobre os objectos não

diferem conforme a vista pelo qual se acede aos mesmos (Ex: Numa vista relativa a um segmento IP é possível aceder a atributos dos objectos relativos a X.25). O mesmo se passa com o tratamento e propagação dos eventos.

A nível da representação de uma rede, o *Tnm::repo* tem a vantagem de adequar as representações dos componentes da rede ao contexto de cada vista. Esta característica verifica-se ao nível dos atributos dos componentes e do tratamento de eventos, permitindo reflectir na representação da rede as múltiplas perspectivas da gestão de uma forma mais adequada aos seus respectivos contextos.

VII. Conclusões

A evolução dos sistemas computacionais, quer em termos do seu crescimento quer em termos da sua heterogeneidade, criou a necessidade de existência de ferramentas de gestão que por um lado, ofereçam uma perspectiva global de toda a infra-estrutura de sistemas de informação alvo da gestão mas que também permitam por outro, segmentar e especializar em vistas parcelares, possibilitando a gestão especializada de sub-sistemas.

O modelo proposto agrega aspectos relevantes que já se podiam encontrar em alguns dos sistemas revistos, embora de forma dispersa e não integrada, e introduz algumas características que se julgam inovadoras. Enumeram-se as características mais relevantes do modelo proposto:

- Separação entre as entidades que representam os elementos da rede e as entidades que representam as múltiplas relações em que estes podem estar envolvidos (característica patente na plataforma *HP OpenView*).
- Modelo de eventos que permite caracterizar o comportamento de cada elemento da rede com diferentes graus de refinamento (inspirado no sistema *Tnm::map*).
- Mecanismo de definição de políticas de acesso aos objectos (patente no sistema Domains).
- A capacidade das vistas filtrarem a informação dos elementos da rede que é visível ao utilizador da vista
- A expansão do modelo de eventos do *Tnm::map* por forma a permitir que a execução de eventos tenha a capacidade de se propagar de uma forma ascendente, descendente e lateral

O esforço de concretização através de uma extensão Tcl, reforçou a nossa convicção de que as linguagens interpretadas com as características do Tcl apresentam vantagens únicas na modelação e prototipagem de ferramentas de gestão de rede que compensam largamente a perca de desempenho que as caracterizam.

Em termos de perspectivas de melhoramentos futuros ao sistema proposto, apontam-se as seguintes direcções:

- O encapsulamento e herança múltipla, características das linguagens orientadas por objectos, são duas propriedades interessantes quando pretendemos

representar a rede. A utilização deste paradigma na constituição de um sistema de vistas vocacionado para a gestão de rede, é sem duvida alguma, uma expansão interessante ao nosso trabalho.

- No *Tnm::repo* as restrições são aplicadas ao nível da vista. Seria interessante generalizar o uso das restrições aos diferentes níveis hierárquicos da vista.
- O mecanismo de restrição embora eficaz é elementar e passível de ser melhorado.

VIII. Referências

[**1**] J. Schönwälder and H. Langendörfer, Technical University of Braunschweig, Germany, Tcl Extensions for Network Management Applications. 1995.

[**2**] John K. Ousterhout, 1990 Winter Usenix Conference Proceedings. Tcl: An Embeddable Command Language. 1989.

[**3**] Brent B. Welch, Prentice Hall PTR. Practical Programming in Tcl and Tk. 1997

[**4**] J. Schönwälder and H. Langendörfer, Technical University of Braunschweig, Germany, INED, An application Independent Network Editor. 1993.

[**5**] Computer Associates, Unicenter TNG SDK Programmer's Guide

[**6**] Hewlett Packard, OpenView Windows Developer's Guide

[**7**] M. Sloman and J. Moffett, "Domain Management for Distributed Systems". 1989.

[**8**] Sethi A.S. Sherwin, C.M. Kalyanasundaram, P. and D.Zhu. A spreadsheet-based scripting environment for snmp. 1997

[**9**] W. Yeong. Snmp Query Language. 1997

[**10**] Nikolaos Anerousis, AT&T Labs Research,An Architecture for Building Scalable, Web-based Management Services.

[**11**] Manual do Scotty , (http://wwwsnmp.cs.utwente.nl/~schoenw/scotty/)

[**12**] Carlos Palma , Faculdade de Ciências, Universidade de Lisboa, Suporte para vistas em ambientes de gestão de rede. 2000.