Esquema de Identificação Uniforme de Recursos SNMP

Rui Pedro Lopes[1], José Luis Oliveira[2]

[1]Instituto Politécnico de Bragança, ESTiG, 5300 Bragança (rlopes@ipb.pt)
[2]Universidade de Aveiro, DET, 3810 Aveiro (jlo@det.ua.pt)

Palavras chave: SNMP, URI, URL, Gestão de Redes

Resumo
Um dos efeitos globalizadores da Web, para além da sua omnipresença em sistemas computacionais, foi a uniformização, numa única interface de acesso, de diversos serviços até então utilizados de forma dispersa por aplicações independentes. Os recursos da Internet passaram a ser identificados por intermédio de um esquema URI, uma cadeia de caracteres com sintaxe e semântica próprias.
Apesar destes identificadores estarem definidos para múltiplos serviços, tais como http, ftp, gopher e news, a sua adopção para SNMP nunca foi equacionada.
Este artigo propõe um esquema URI para identificar recursos SNMPv1, SNMPv2c e SNMPv3 e apresenta ainda diversos cenários onde a existência de um mecanismo de identificação e de localização compacto e completo como este acrescenta grande funcionalidade às aplicações de gestão de redes.

I. Introdução

A informação, os serviços ou, genericamente, qualquer dos recursos que diariamente se acrescentam à Internet tem subjacente um espaço de nomes, parcialmente baseado no DNS. A identificação de recursos é efectuada por linhas de texto denominadas URI (*Uniform Resource Identifiers*) [1]. Esta cadeia de caracteres concentra a informação necessária para consultar ou, eventualmente, modificar a informação contida num determinado recurso.

Os recursos podem ser físicos, como um processador, memória ou dispositivos de armazenamento de massa, ou lógicos, como um servidor de páginas de Internet ou um agente de gestão de redes. Os recursos são localizados por intermédio de referências que assinalam a sua localização e natureza. Por exemplo, a caixa de correio electrónico de um determinado utilizador é assinalada segundo o esquema `mailto:<nome>@<endereço>`.

Apesar das diferenças intrínsecas a cada recurso, a sua referenciação é regida por conceitos comuns como o nome ou o endereço. Este conjunto de características permite utilizar uma sintaxe uniforme para os referenciar independentemente da natureza dos recursos.

A uniformidade de representação de referências permite utilizar identificadores para recursos diferentes num mesmo contexto. No contexto da Internet, por exemplo, os

recursos FTP, HTTP ou NEWS são acessíveis por intermédio de uma ferramenta comum – o navegador de Internet. O recurso é especificado no campo de endereço por intermédio do URI e de acordo com a sua semântica é invocada a ferramenta adequada para o seu processamento e apresentação.

A gestão de redes é tradicionalmente associada ao modelo SNMP (*Simple Network Management Protocol*) [2]. Para gerir recursos SNMP utiliza-se normalmente uma estação de gestão que emite comandos e recebe notificações SNMP. Estes serviços são tipicamente disponibilizados através de APIs especialmente construídas e que fazem depender a sua utilização de um conjunto de argumentos que permitem identificar a informação de gestão e a acção a executar sobre essa informação. Para além do problema que é diferentes API disponibilizarem diferentes e incompatíveis interfaces, devemos ainda preocupar-nos com a versão utilizada, que implica argumentos diferenciados. Tal como para o contexto Internet é importante integrar as diferentes ferramentas, paradigmas e modelos em torno de um contexto comum.

Neste artigo propomos a utilização de um esquema URI específico para SNMP que resolve facilmente a dispersão de argumentos associados a operações de gestão. Seguindo esta abordagem, a gestão SNMP pode ser efectuada com base numa única ferramenta, onde os diferentes recursos e operações são identificados pela semântica de um URI dedicado.

O conceito de URI representativo de recursos SNMP tem vindo a ser utilizado pelos autores, de forma mais ou menos informal, em ferramentas gráficas para gestão de plataformas de agentes móveis. Estas podem ser acedidas por CORBA ou SNMP e um dos requisitos é suportar ambos os protocolos. Para lidar com os diferentes esquemas, foram utilizados URL específicos, o que permitem distinguir entre as diferentes formas de acesso [3].

Recentemente, surgiram também dois projectos de gestão de redes que partilham deste conceito. Um deles, *iosnmp* [4], consiste num *plugin* para o sistema gráfico KDE [5] que, após registo, permite utilizar o *konqueror* como *browser* de MIBs juntamente com o sistema de ficheiros local ou páginas HTML. De acordo com os autores, este módulo aceita URI da forma `snmp://v3user@host:port/initialMibNode/` e suporta unicamente o SNMPv3. Como parâmetros de segurança, assume o nível de segurança `authNoPriv` e pede a chave de acesso ao utilizador numa janela própria. Outros parâmetros, como o contexto, *timeout*, *retries* ou outros não são definidos. A aplicação *Cricket* é outro exemplo e consiste numa ferramenta de monitorização de recursos. Estes podem ser configurados por intermédio do URL adequado, do tipo `snmp://comunidade@host:port/fonte`, onde fonte representa o objecto de gestão [6]. Esta proposta prevê apenas as versões SNMPv1 e SNMPv2c e, tal como o exemplo anterior, não admite outro tipo de informação associada ao protocolo como *timeout*, *retries* ou outros.

Apesar das limitações individuais, estes exemplos suportam a necessidade de existência de um URL que descreva recursos SNMP independentemente da versão, parâmetros de segurança e operações de forma uniforme e transversalmente ao tipo de produto. Tal como acontece com os *browsers* de Internet, o URL não muda apesar do programa poder ser diferente.

Por outro lado, estas três utilizações de URLs [3][4][6] são baseadas exclusivamente no gestor, como método de acesso à informação de gestão. No presente artigo, procuramos que o formalismo de URI para SNMP seja estendido ao lado do agente

como um sistema de identificação mais poderoso do que o que é fornecido pelo sistema de OIDs para referenciar atributos dentro de uma MIB.

Este artigo descreve, na secção II, a sintaxe e a semântica necessárias para identificar recursos SNMP num formato compatível com URI, a que designamos SNMP URL. Na secção III são apresentados alguns cenários de utilização prática deste conceito. No primeiro baseamos-nos nos SNMP URL para propor uma URL-TARGET-MIB em substituição da SNMP-TARGET-MIB. No segundo apresentamos a sua utilização em gestores distribuídos para dar possibilidade de gestão remota ao módulo Expression MIB [7], actualmente em investigação. O terceiro incide sobre a sua utilização, do lado do gestor, no caso prático de gestão de agentes móveis [3].

II. Localização Uniforme de Recursos SNMP

O conceito URI foi definido pelo grupo de redes do IETF (*Internet Engineering Task Force*) como a especificação de referências universais para recursos físicos ou lógicos [1]. A sua sintaxe é suficientemente genérica para que possa ser aplicado na identificação de um grande conjunto de recursos, nomeadamente, páginas de Internet, endereços de correio electrónico, grupos de discussão e livros, entre muitos outros. No âmbito do IANA existem actualmente esquemas de URL para 36 serviços [8].

A. Sintaxe Genérica dos URI

No seu nível mais abstracto um URI tem o seguinte formato:

```
[esquema:]parte-dependente-do-esquema[#fragmento]
```

Os símbolos '[' e ']' indicam secções opcionais enquanto que os símbolos ':' e '#' delimitam as várias secções. Os URI dizem-se absolutos, quando é indicado o seu esquema, e relativos no caso contrário. Neste último caso, o URI relativo é construído com base num URI absoluto, de forma a completar a informação em falta.

Os URI podem também ser classificados como hierárquicos. Neste caso, a parte dependente do esquema começa com o símbolo '/' e tem o seguinte formato:

```
[esquema:][//autoridade][caminho][?consulta][#fragmento]
```

A autoridade representa o topo hierárquico de um esquema de identificação, como `[utilizador@]endereço[:porto]` (de notar que o símbolo '/' precede a autoridade no formato acima). O `caminho` contém dados que identificam um recurso único no contexto da autoridade em causa. A secção `consulta` contém informação que irá ser entregue e interpretada pelo recurso. O `fragmento` consiste na informação adicional a ser interpretada do lado do utilizador após a operação de consulta ser realizada com sucesso. Teoricamente não faz parte do URI, uma vez que não é comunicado ao recurso, mas é frequentemente associado (para controlo de marcadores em páginas html, por exemplo).

Os URI urn:isbn:096139210x, que assinala um livro, e mailto:pemz@mail.hosting.pt, que especifica um endereço de correio electrónico, são não hierárquicos ou opacos, de acordo com a nomenclatura definida nas normas.

O URI http://www.ics.uci.edu/pub/ietf/uri/#Related é absoluto e hierárquico, uma vez assinala o esquema http e a parte dependente do esquema começa com o símbolo '/'. A autoridade representa um nome de Internet (www.ics.uci.edu) e o caminho assinala o recurso /pub/ietf/uri/, único no âmbito desse servidor. O fragmento, utilizado nesta

caso pelo navegador de Internet para deslizar a página verticalmente, não é comunicado ao servidor e contém a informação "Related".

A Tabela 1 apresenta alguns exemplos de URI já normalizados.

Tabela 1 – Exemplos de URIs.

Modelo	URI
HTTP	http://www.det.ua.pt
FTP	ftp://jprs@ftp.univ.pt/private/projectX/
XMLORG	urn:xmlorg:objects:schema:xmlschema:xcatalog
NFS	nfs://server/a/b
LDAP	ldap://ldap.itd.umich.edu/o=University%20of%20Michigan,c=US
MAIL	mailto:rlopes@ipb.pt

B. Parâmetros SNMP

A funcionalidade de gestão associada às entidades SNMP, de acordo com a especificação da versão SNMPv3, é definida por agrupamento de aplicações. Estas podem ser de quatro tipos: geradoras de comandos, geradoras de notificações, receptoras de comandos e receptoras de notificações. Como tal, um agente de gestão SNMP é constituído por um gerador de notificações e um receptor de comandos enquanto que as aplicações de gestão são constituídas por um receptor de notificações e um gerador de comandos.

As entidades SNMP são identificadas univocamente por um número designado por `snmpEngineID`. Cada entidade poderá conter diversos contextos, únicos em cada entidade. Para identificar cada objecto são, consequentemente, necessários quatro parâmetros: o identificador do mecanismo SNMPv3 (`snmpEngineID`), o nome do contexto (`contextName`), o identificador de objecto (OID – ex. `ifDescr`) e o identificador de instância (ex. '1'). No caso do SNMPv1, a consulta é efectuada apenas indicando o identificador de objecto e o identificador de instância, o que resulta numa maior simplicidade de acesso mas também numa menor flexibilidade.

A comunicação entre entidades SNMP rege-se por um conjunto de parâmetros que asseguram a correcta identificação dos intervenientes, a segurança da comunicação e o processamento das mensagens. Um dos aspectos fundamentais é a segurança. De acordo com a versão do protocolo podem ser utilizados vários modelos de segurança. Para as versões SNMPv1 e SNMPv2c, a privacidade é inexistente e a autenticação é efectuada com base num nome de comunidade. Para a versão SNMPv3, existe já a possibilidade de encriptação das mensagens, o que garante a privacidade da comunicação, e modelos de autenticação com base em nomes de utilizadores e correspondentes palavras chave.

Resumidamente, para que a comunicação entre entidades SNMP seja possível, é necessário indicar o protocolo a ser utilizado. Este consiste actualmente numa de três versões: SNMPv1, SNMPv2c ou SNMPv3. Associado ao protocolo encontra-se, obrigatoriamente, o endereço do destinatário bem como os parâmetros relacionados com a segurança. Estes últimos poderão consistir numa cadeia de caracteres representativa de uma comunidade para as versões SNMPv1 e SNMPv2c ou em três parâmetros para o SNMPv3:

- Modelo de segurança – SNMPv1, SNMPv2c, USM (*User Security Model*).
- Nome de utilizador – cadeia de caracteres.
- Nível de segurança – sem autenticação nem privacidade (`noAuthNoPriv`), com autenticação e sem privacidade (`authNoPriv`), com autenticação e privacidade (`authPriv`).

Adicionalmente, será necessário passar ao módulo de segurança as chaves associadas aos algoritmos de autenticação e de privacidade.

O protocolo poderá também receber indicação do tempo que deverá esperar por uma resposta (*timeout*) e do número de tentativas que deverá efectuar antes de desistir de contactar o destinatário (*retry count*)

Finalmente, para aceder ao recurso pretendido, ou seja, à informação de gestão, será necessário assinalar o contexto, OID (Object ID) e instância.

C. URL para SNMP

Um SNMP URL apresenta toda a informação necessária para a comunicação entre entidades SNMP num formato compatível com o conceito URI.

De acordo com a especificação, os URI têm restrições em termos de símbolos ou caracteres que, devido à sua extensão não são apresentados neste documento. Para mais detalhes refira-se à consulta de [1].

Genericamente, a sintaxe proposta é do tipo:

```
snmpurl      =  esquema "://" [segurança "@"] [end_porto] ["/"
                [recurso] ["?" [operação] ["?" [versão] ["?"
                [contexto]]]]] ["#" parâmetros]
esquema      =  "snmp"
segurança    =  [comunidade] |
                [utilizador [":" autenticação [":" privacidade]]]
comunidade   =  comunidade da secção 3.2.5 de [9]
utilizador   =  nome de utilizador da secção 2.1 de [10]
autenticação =  elemento_a ?("&" elemento_a)
elemento_a   =  "auth=" protocolo_a | "pass=" chave
privacidade  =  elemento_p ?("&" elemento_p)
elemento_p   =  "priv=" protocolo_p | "pass=" chave
protocolo_a  =  protocolo de autenticação da secção 1.4.2 de [10]
protocolo_p  =  protocolo de privacidade da secção 1.4.3 de [10]
chave        =  cadeia de caracteres
end_porto    =  endereço [":" porto]
endereço     =  endereço IP ou nome associado
porto        =  número inteiro
recurso      =  OID ["/" instância]
OID          =  sequência de inteiros separados por '.' |
                nome associado
instância    =  sequência de inteiros separados por '.'
operação     =  "op=" nome_op [op_params]
nome_op      =  nome da operação SNMP. Actualmente
                ("get" | "getNext" | "set" | "trap" | "response" |
                "getBulk" | "inform" | "trap2")
op_params    =  op_param *("&" op_param)
op_param     =  ("value=" valor | "maxrep=" valor | "nonrep=" valor)
valor        =  cadeia de caracteres
versão       =  "v1" | "v2c" | "v3"
contexto     =  contexto da secção 3.3.1 de [11]
```

```
parâmetros    =  parâmetro *("&" parametro)
parâmetro     =  "timeout=" inteiro | "retries=" inteiro
```

Exemplos de SNMP URL para a versão 1 são:

snmp://private@sw1.estig.ipb.pt/sysContact/0?op=set&value=Rui%20Lopes

snmp://public@nms.estig.ipb.pt:161/sysUpTime?op=getNext

Para SNMPv2c serão do tipo:

snmp://private@sw1.estig.ipb.pt/sysContact/0?op=set&value=Rui%20Lopes?v2c

snmp://public@nms.estig.ipb.pt:161/sysUpTime?op=getNext?v2c

E para SNMPv3:

snmp://rlopes@sw1.estig.ipb.pt/sysContact/0?op=set&value=Rui%20Lopes?v3

snmp://guest@nms.estig.ipb.pt:161/sysUpTime?op=getNext?v3?router

O prefixo '%' é utilizado para representar caracteres especiais. A combinação "%20" corresponde ao espaço.

De notar que apesar de ser possível indicar chaves e parâmetros de segurança directamente no URL este procedimento não é recomendado devido às falhas que poderão resultar da utilização de sequências de caracteres "em claro". Este inconveniente pode ser resolvido através de interacção com o utilizador (a aplicação apresenta uma janela específica para a introdução de informação de segurança) ou, no caso de não haver interacção com o utilizador, armazena a informação de segurança de forma ilegível (cifrada) em ficheiros associados de configuração. Esta última opção é actualmente utilizada em algumas aplicações SNMPv3 no papel de agentes [12].

III. Aplicações Práticas

Para melhor compreender os conceitos anteriores e para ilustrar alguns casos, apresentam-se de seguida algumas aplicações práticas. A primeira aplicação óbvia, que resulta desta proposta, é a integração de um MIB *browser* directamente num *browser* Web. Os exemplos seguintes privilegiam a utilização dos SNMP URL do lado dos agentes.

A. URL-TARGET-MIB

As aplicações SNMP podem, por vezes, necessitar contactar outras aplicações possivelmente remotas para obter algum parâmetro ou enviar alguma notificação. A arquitectura SNMP define um módulo denominado SNMP-TARGET-MIB, com a finalidade de agrupar os parâmetros necessários à comunicação em torno de listas de nomes [13]. A aplicação armazena apenas o nome pretendido e obtém os parâmetros através de consultas sobre a SNMP-TARGET-MIB.

Resumidamente, a SNMP-TARGET-MIB permite associar listas de nomes a um conjunto de parâmetros que especificam o endereço, porto, protocolo e os mecanismos de segurança ligados ao protocolo. Por exemplo, permite armazenar a seguinte informação: para contactar o "router1" ou a "coreBridge" utilizar o protocolo=SNMPv3, *timeout*=5, *retry count*=3, utilizador="senior", modelo de segurança="USM", nível de segurança="authNoPriv". Estes parâmetros são referenciados a partir de outras MIB por intermédio do nome, neste caso "router1" ou "coreBridge".

Outra abordagem aqui proposta é associar nomes a entradas do tipo URL numa tabela, de forma a indicar apenas os nomes que serão resolvidos para valores, obtidos por intermédio do URL associado. Esta abordagem, que designamos por URL-TARGET-MIB, permite complementar ou substituir o módulo SNMP-TARGET MIB (Figura 1).

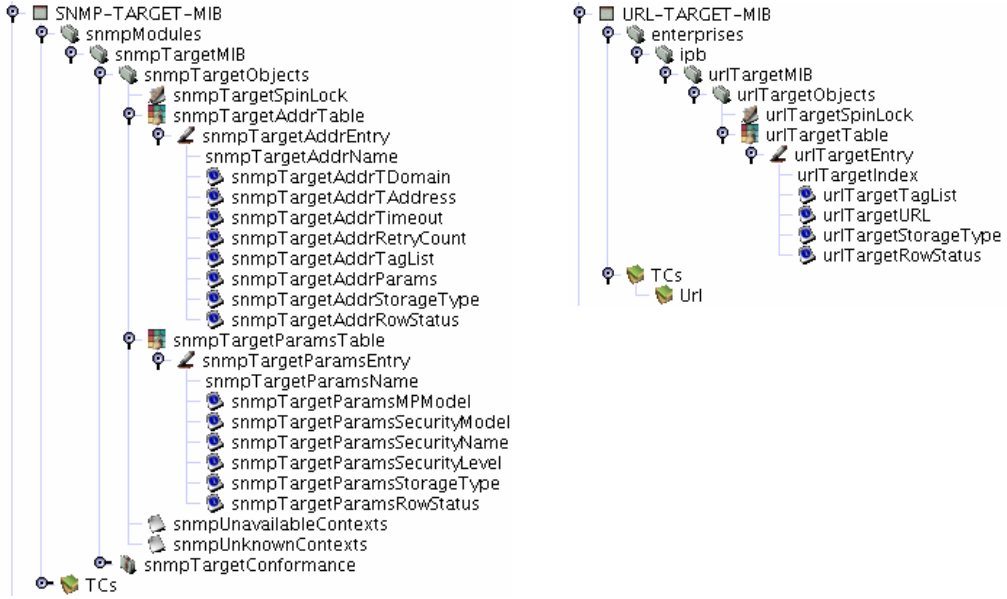

Figura 1 – Comparação entre a SNMP-TARGET-MIB e a URL-TARGET-MIB.

A maior diferença entre as duas abordagens é a extensão do módulo, resultando num agente mais simples. Outra vantagem é a possibilidade de armazenar também OIDs na mesma MIB, o que não acontece com a SNMP-TARGET-MIB.

B. Modificações sobre a Expression MIB

O grupo de trabalho DISMAN definiu, no âmbito do IETF [14], um conjunto de MIB com o objectivo de descentralizar, num conjunto de gestores intermédios, as tarefas tipicamente associadas à estação de gestão central. Do ponto de vista dos outros agentes estes assumem o papel de gestor enquanto que do ponto de vista das aplicações de gestão apresentam o comportamento de agente. Neste caso, a instrumentação é efectuada directamente sobre as operações de gestão. A arquitectura permite criar um conjunto de "ilhas" de gestão hierárquicas de forma a tornar o sistema mais robusto pela introdução de redundância, mais escalável e permitir operação em situações de interrupção de conectividade.

Um dos módulos definidos pelo grupo é a Expression MIB. A Expression MIB foi criada com o propósito de tornar possível a definição de objectos que não foram inicialmente considerados durante a definição de outros módulos MIB [15]. Por outras palavras, permite especificar expressões baseadas em objectos de gestão existentes. Permite também encadear expressões, ou seja, definir expressões cujos parâmetros dependam do resultado de outras expressões.

Uma expressão é composta por operadores, funções e valores. Os valores podem ser constantes ou variáveis estando associados a OIDs específicos. A expressão é definida por uma cadeia de caracteres onde se especificam todos os seus componentes.

É precisamente na utilização de variáveis que este módulo concentra simultaneamente o seu poder e o seu inconveniente. Da forma como foi definida, a MIB não permite obter valores provenientes de agentes remotos, estando confinada à definição de expressões com valores inteiramente locais. Este facto limita o número de expressões possíveis e impede o correlacionamento de informação com proveniências distintas.

As variáveis são assinaladas numa tabela que apenas contém a coluna correspondente ao OID, faltando os outros parâmetros necessários para contactar os agentes remotos, nomeadamente, endereço, porto, protocolo, parâmetros de segurança e contexto. Qualquer expressão pode ser representada no seguinte formato genérico:

$$x = Expression(oid_1, oid_2, ... oid_n)$$

A utilização de URL em vez de OIDs na coluna atrás referida permite, com um aumento mínimo de complexidade, dotar a Expression MIB com a possibilidade de consulta remota de valores independentemente da origem aumentando a sua flexibilidade e capacidade. Passamos a poder contar com expressões do tipo:

$$x = Expression(url_1, url_2, ... url_n)$$

Esta associação pode ser efectuada de duas formas. A primeira passa pela adopção de uma MIB específica para armazenar e indexar os URL. Desta forma, as os argumentos da expressão podem ser definidos como referências para URL. A vantagem desta abordagem é a compatibilidade com a SNMP-TARGET-MIB, definida para o SNMPv3. Por outro lado, o inconveniente é a necessidade de associar mais um módulo MIB ao agente. Outra forma, mais simples, passa pela associação directa do URL à variável. Neste caso, o armazenamento de OID é substituído pelo armazenamento de URL o que implica modificar apenas o tipo de dados da coluna associada para texto.

C. Gestão de agentes móveis

Os agentes móveis são instâncias de um paradigma mais abrangente denominado agentes de *software*. Estes consistem em programas que desempenham determinada função em benefício de outro programa ou do utilizador. Um agente com capacidade de se deslocar ao longo da rede é designado por agente móvel. Estes podem, portanto, ser criados em qualquer ponto da rede, interromper temporariamente a execução, deslocar o código e o estado e retomar a execução num outro ponto da rede.

Este tipo de programas necessitam de plataformas designadas por agência para lhes conceder o ambiente de execução. Há, no entanto, problemas de interoperabilidade se os agentes pretendem deslocar-se entre agências de diferentes fabricantes. A interoperabilidade entre ambientes de execução encontra-se em desenvolvimento pelo OMG sob a forma de um documento formal – MAF [16]. Esta especificação tem como objectivo permitir que um agente móvel possa deslocar-se entre agências com perfil semelhante (partilhando a mesma linguagem, tipo de agência, tipo de autenticação e método de serialização) através de um conjunto de interfaces CORBA normalizadas. Além disso, permite também que os agentes possam ser geridos através da monitorização e actuação sobre o seu ciclo de vida.

A gestão de agentes móveis é fundamentalmente efectuada através da agência por intermédio de mecanismos proprietários ou, caso seja possível, por intermédio das interfaces definidas na arquitectura MAF. Em trabalho desenvolvido anteriormente, foi definida uma MIB para converter comandos SNMP em invocações MAF o que

torna possível gerir plataformas de agentes móveis por SNMP ou por MAF em simultâneo [17].

Neste contexto, foi desenvolvida uma ferramenta gráfica de gestão de agentes móveis que suporta as duas formas de acesso numa interface comum, apesar do método de acesso ser substancialmente diferente: SNMP ou CORBA (Figura 2).

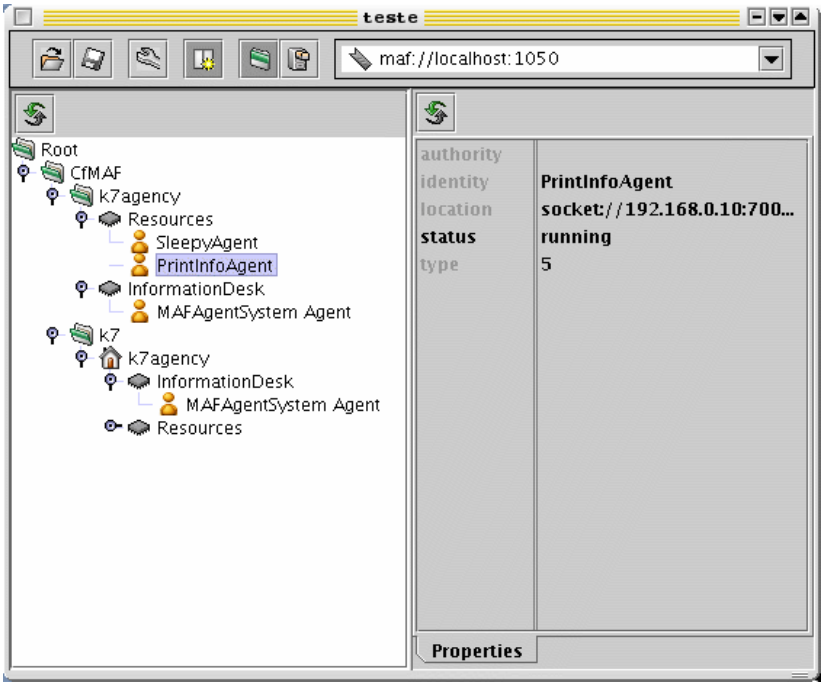

Figura 2 – Ferramenta de gestão baseada em URI.

De forma semelhante à definição de SNMP URL, foi definido um URL para MAF, do tipo maf://<endereço do servidor de nomes>:<porto>/<contexto>. A sintaxe semelhante, apesar das diferenças de semântica, permite definir uma lista uniforme de referências (*bookmarks* ou lista de favoritos) independentemente do modelo de gestão. De acordo com o URL indicado, a ferramenta gráfica carrega o módulo associado e procede de acordo com os comandos emitidos pelo utilizador.

A figura anterior apresenta uma árvore de recursos relacionados com agentes móveis e foi construída após indicar o URL no campo de endereço (canto superior esquerdo). Caso seja indicado um SNMP URL, a árvore é modificada de acordo com a informação proveniente do agente SNMP.

IV. Conclusões

Os conceitos de URI e URL, apesar de largamente utilizados como formatos de identificação e localização na Internet, têm estado afastados da gestão de redes e sistemas.

Este artigo propõe um esquema URI para identificar recursos SNMPv1, SNMPv2c e SNMPv3. Esta abordagem, designada por SNMP URL, permite especificar numa única linha de texto todos os parâmetros necessários à comunicação SNMP, independentemente da versão e do modelo de segurança.

Para validar o SNMP URL são apresentados alguns casos práticos. O primeiro exemplo é uma proposta alternativa à SNMP-TARGET-MIB e que usa URL para identificar objectos de gestão locais ou remotos. Outro exemplo, é uma proposta de extensão à Expression MIB que permite o processamento de expressões matemáticas sobre qualquer conjunto de objectos numéricos obtidos de múltiplos agentes. É ainda sugerida a utilização de SNMP URL em aplicações gestoras permitindo distinguir o tipo de serviço a invocar, à semelhança do que acontece com os *browsers* de Internet.

V. Referências

[1] T. Berners-Lee, R. Fielding, U.C. Irvine, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", *Internet Request for Comments 2396*, Agosto 1998.

[2] J. Case, R. Mundy, D. Partain, B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", *Internet Request for Comments 2570*, Abril 1999.

[3] R. Lopes, J. Oliveira, "SNMP Management of MASIF Platforms", *actas do IFIP/IEEE International Symposium on Integrated Management – IM'2001*, Seattle, Maio 2001.

[4] iosnmp (http://www.opensnmp.org/).

[5] KDE (http://www.kde.org/).

[6] Cricket (http://cricket.sourceforge.net/).

[7] R. Lopes, J. Oliveira, "Distributed Management: Implementation issues", *actas da International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet – SSGRR'2000*, l'Aquila, Roma, Itália, Agosto 2000.

[8] Uniform Resource Identifier (URI) SCHEMES (http://www.iana.org/assignments/uri-schemes).

[9] J. Case, M. Fedor, M. Schoffstall, J. Davin, "A Simple Network Management Protocol (SNMP)", *Internet Request for Comments 1157*, Maio 1990.

[10] U. Blumenthal, B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", *Internet Request for Comments 2574*, Abril 1999.

[11] D. Harrington, R. Presuhn, B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", *Internet Request for Comments 2571*, Abril 1999.

[12] The NET-SNMP Project (http://www.net-snmp.org/).

[13] D. Levi, P. Meyer, B. Stewart, "SNMP Applications", *Internet Request for Comments 2573*, Abril 1999.

[14] DISMAN Charter (http://www.ietf.org/html.charters/disman-charter.html).

[15] R. Kavasseri, B. Stewart, "Distributed Management Expression MIB", *Internet Request for Comments 2982*, Outubro 2000.

[16] Mobile Agent Facility Specification, Object Management Group, 00-01-02.pdf (ftp://ftp.omg.org/pub/docs/formal/00-01-02.pdf).

[17] R. Lopes, J. Oliveira, "Descrição e Implementação de uma MIB para Sistemas MASIF", *actas da 3ª Conferência de Redes de Computadores – CRC2000*, Évora, Novembro 2000.