

Mestrado em Engenharia Informática

# **Sistemas de Designação Distribuidos**

*João Alexandre Carvalho Pinheiro Leite*

*Fevereiro 1995*

---

<b>INTRODUÇÃO</b>	<b>3</b>
<hr/>	
<b>DESIGNAÇÃO E SISTEMAS DISTRIBUIDOS</b>	<b>4</b>
<hr/>	
NOMES E ATRIBUTOS	4
NOMES NOS SISTEMAS DISTRIBUIDOS	4
SERVIÇOS DE DESIGNAÇÃO DISTRIBUIDOS	4
REQUISITOS DE UM SERVIÇO DE DESIGNAÇÃO	5
COMPONENTES DE UM SERVIÇO DE DESIGNAÇÃO	5
RESOLUÇÃO DE NOMES E NAVEGAÇÃO	5
GESTÃO INTEGRADA DE NOMES	6
DESIGNAÇÃO BASEADA EM ATRIBUTOS	6
A NOÇÃO DE GRUPO	7
<hr/>	
<b>DNS-DOMAIN NAME SERVICE</b>	<b>8</b>
<hr/>	
INTRODUÇÃO	8
ESPAÇO DE DESIGNAÇÃO	8
SERVIDORES DE NOMES	9
RESOLVERS	10
COMENTÁRIOS	11
<hr/>	
<b>GNS-GLOBAL NAME SERVICE</b>	<b>12</b>
<hr/>	
INTRODUÇÃO	12
CLIENTE	12
ADMINISTRADOR	14
ESPAÇO DE DESIGNAÇÃO	15
COMENTÁRIOS	17
<hr/>	
<b>X500-DIRECTORY SERVICE</b>	<b>18</b>
<hr/>	
INTRODUÇÃO	18
O MODELO DE INFORMAÇÃO	18
O MODELO FUNCIONAL	19
COMENTÁRIOS	21
<hr/>	
<b>CONCLUSÕES</b>	<b>23</b>
<hr/>	
<b>BIBLIOGRAFIA</b>	<b>24</b>
<hr/>	

## INTRODUÇÃO

---

Um qualquer sistema informático é composto por um conjunto de elementos que interagem e cooperam com vista a alcançar um determinado fim. Todos esses componentes devem poder ser referidos de uma forma consistente, quer pelo sistema quer pelo utilizador por forma a serem manipulados e partilhados. Num sistema distribuído, a tarefa de designar os seus componentes torna-se obviamente mais complexa devido ao tamanho do próprio sistema e acima de tudo à necessidade de transparência no que se refere à localização dos componentes.

Os serviços de designação são componentes do sistema que quando invocados por um cliente, obtêm os atributos de um dado objecto dado o seu nome.

O capítulo seguinte aborda brevemente alguns assuntos relacionados com a designação, com particular relevo para a sua aplicação aos sistemas distribuídos. Não pretendendo cobrir exhaustivamente todos os tópicos relacionados com este tema, tem apenas como objectivo uma introdução aos restantes capítulos onde serão descritas três abordagens propostas: o *Domain Name Service* (DNS), o *Global Name Service* (GNS) e o *X.500-Directory Service*.

## DESIGNAÇÃO E SISTEMAS DISTRIBUIDOS

---

### Nomes e atributos

Uma entidade que necessita aceder a um recurso tem de possuir um nome que o refira.

Num qualquer sistema distinguem-se três tipos de nomes:

- nomes externos ou simbólicos: normalmente cadeias de caracteres de fácil manipulação por parte dos humanos;
- nomes internos globais: identificadores escolhidos por forma a aumentar a eficiência da sua manipulação por parte do sistema. São normalmente sequências de bits de comprimento fixo;
- nomes internos contextuais ou locais: camada intermédia de nomes entre os dois tipos anteriores. Esta categoria existe em alguns sistemas com o objectivo de esconder os nomes internos globais por questões de segurança.

Existe um mapeamento entre os três tipos acima referidos e daqui em diante serão referidos apenas por nomes.

A associação entre um nome e o objecto a que ele se refere é chamado um *binding*. Os nomes específicos a um serviço estão associados (*bound*) pelo serviço à representação actual do objecto ou recurso em causa. Os nomes que se referem a entidades para além do espectro de um único serviço (utilizadores, computadores, serviços, etc) estão associados (*bound*) a atributos do objecto designado. Entre os atributos podemos ter desde *passwords*, números de telefone, mas sem dúvida os atributos mais importantes são os endereços.

### Nomes nos sistemas distribuídos

As camadas de nomes anteriormente descritas estão igualmente presentes nos sistemas distribuídos. Este tipo de sistemas, pelas suas características, apresenta alguns problemas específicos nomeadamente no que respeita às características dos nomes utilizados. Em particular, um nome interno global deve ser independente da máquina onde o objecto reside actualmente por forma a permitir a sua migração. Por outro lado é necessária uma forma de garantir a unicidade deste tipo de nomes de uma forma descentralizada. Estes problemas são resolvidos através da utilização de nomes internos únicos e globais (UID) que são caracterizados pelas seguintes propriedades:

- unicidade no espaço: se um processo conhece o UID de um objecto, não tem de memorizar a sua localização. Isto permite a migração de objectos sem perda de identidade, garantindo também o significado de um nome quando utilizado por outra entidade noutro local. O sistema deve ser capaz de encontrar o objecto a partir do UID;
- unicidade no tempo: um UID não deve gerar ambiguidades ao longo do tempo. Quando um nome é utilizado, ou o objecto que ele designa existe, ou o objecto não existe e esse nome não poderá designar outro objecto.

A geração de UID's tem de ser feita de uma forma descentralizada. Para se obter as características desejadas, o método mais comum é a concatenação de uma parte que identifica de forma única o local de criação do objecto com uma parte única dentro desse local.

### Serviços de designação distribuídos

Um serviço de designação mantém uma base de dados com associações (*bindings*) entre um conjunto de nomes e os seus atributos para objectos como por exemplo utilizadores,

computadores que existem fora de qualquer serviço específico. A operação principal que um serviço de designação suporta é a de resolver um nome i.e., ver na base de dados os atributos de um dado nome. Obviamente serão também necessárias operações para criar e apagar elementos dessa base de dados.

Num sistema distribuído este serviço terá obrigatoriamente de ser descentralizado sob pena de se tornar num *bottleneck*.

## Requisitos de um serviço de designação

Por forma a obter um bom desempenho, um serviço de designação distribuído deve obedecer aos seguintes requisitos:

- ser visto como sendo um espaço global de identificação em vez de um espaço de identificação de computadores contendo objectos localmente identificados;
- gerir um número arbitrário de nomes e servir um número igualmente arbitrário de organizações administrativas;
- possuir um tempo de vida elevado, suportando eventuais alterações na estrutura do espaço de designação;
- ter uma disponibilidade elevada pois a maioria dos outros sistemas depende do seu funcionamento;
- isolar possíveis falhas locais não deixando que elas afectem todo o serviço;
- tolerar entidades suspeitas pois num sistema distribuído não existem componentes confiadas por todos os clientes;
- ser consistente a longo prazo, ou seja, embora permitindo estados transitórios inconsistentes, deve tender para um estado consistente.

## Componentes de um serviço de designação

Do ponto de vista estrutural o serviço de designação é constituído com base em entidades activas chamadas servidores de nomes.

No caso distribuído, os servidores de nomes repartem entre si o espaço de designação e, em cooperação, gerem e fornecem o serviço de designação.

A localização e distribuição dos servidores de nomes deve ser transparente para os clientes do serviço. Isto é conseguido através da utilização de *user agents* (UA) que servem de interface entre o cliente e os servidores de nomes.

## Resolução de nomes e navegação

A resolução de um nome é o processo pelo qual dado um nome se obtêm os atributos a si associados.

O processo pelo qual a resolução de nomes é feita e que envolve a interacção do *user agent* (UA) e dos servidores de nomes (NS), nomeadamente na procura do NS que contém a informação desejada é designado por navegação. Convém recordar o facto de que a base de dados está distribuída pelos diversos NS's, contendo cada um uma parte de todo o espaço de designação. O processo de navegação inicia-se quando um cliente invoca os serviços do UA com um pedido de resolução de um nome.

Existem quatro modelos básicos para a navegação (ver fig. 1):

- navegação iterativa: segundo este modelo, o UA contacta sucessivamente NS's que respondem ou com a resolução do nome ou com uma referencia para outro NS que de alguma forma esteja 'mais perto' da informação requerida;
- navegação recursiva: o UA contacta um NS que, caso não contenha a informação necessária à resolução do nome, age como se fosse um UA contactando outro NS, e assim sucessivamente;
- navegação transitiva: o UA contacta um NS que se não contiver a informação necessária redirecciona o pedido para outro NS. O NS que contiver a resolução do nome responde directamente ao UA. Este método em geral não é utilizado pois torna muito difícil a detecção de falhas nalgum dos componentes intermédios do processo, por parte do UA;
- navegação baseada em *multicast*: o UA envia o pedido a todos os NS's sob a forma de *multicast*, respondendo aquele que contiver a informação necessária. Este método tem o problema de não existirem actualmente algoritmos eficientes de *multicast* em larga escala. Um outro problema sucederia no caso de não existir o nome a resolver: o UA não receberia nenhuma resposta, não podendo distinguir entre uma demora normal no processo ou a inexistencia do nome indicado.

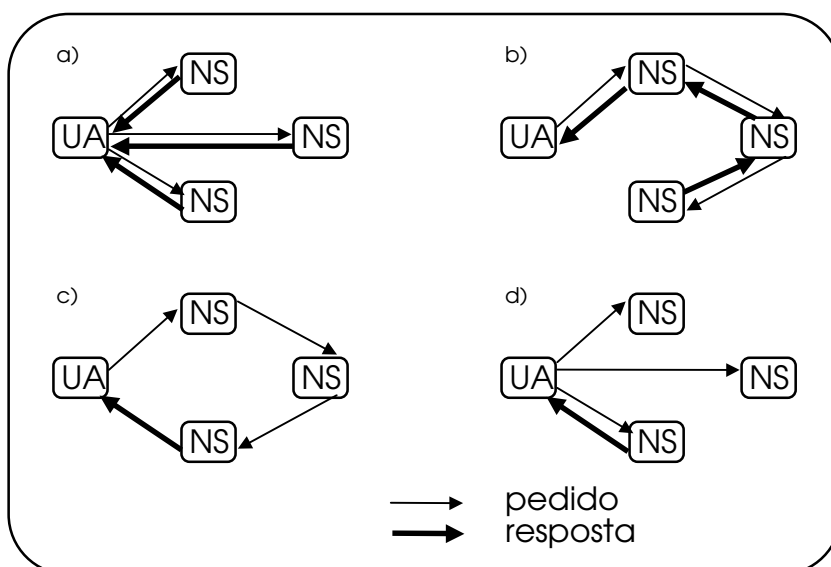


fig.1 - métodos de navegação: a)iterativa b)recursiva c)transitiva d)baseada em multicast

## Gestão integrada de nomes

No *V-System*, sistema distribuido apresentado por Cheriton and Mann, em vez de existir um serviço de designação independente, a gestão dos nomes é feita pelo próprio gestor do objecto.

A justificação utilizada reside no custo envolvido na operação suplementar que seria a de resolver o nome antes de aceder ao objecto. Assim, é proposta uma solução baseada em *multicast*. Para além das desvantagens que este método apresenta e que já foram referidas, este método não é utilizável para designar por exemplo utilizadores. Seria necessário um serviço especial para gerir este tipo de entidades.

## Designação baseada em atributos

Até agora considerámos que um serviço de designação mantém uma base de dados contendo pares do tipo <nome,atributos>. Uma outra aproximação consiste em considerar um sistema de designação em que os atributos servem como chaves de pesquisa. Segundo esta abordagem, o nome é apenas mais um atributo.

Este tipo de serviços são normalmente chamados 'páginas amarelas', e os serviços convencionais 'páginas brancas' por analogia com as listas telefónicas.

Entre as vantagens deste tipo de serviço destaca-se a sua elevada potencia no que se refere a possibilitar a selecção de objectos com certas características apesar de não se conhecer os seus nomes, nomeadamente aquando da escrita de programas e na selecção de serviços.

Como grande desvantagem da designação baseada em atributos está a simplicidade da utilização convencional dos nomes.

O X.500, que será descrito noutro capítulo, é um exemplo deste tipo de abordagem.

## **A noção de grupo**

Num sistema distribuido, por forma a obter uma melhor performance, deve ser permitida a existencia de várias cópias de um dado objecto.

Do ponto de vista do cliente, esse objecto é único e deve ter um nome único (interno ou simbólico) que o designa. Esta unicidade é no entanto apenas lógica.

Do ponto de vista do gestor do objecto, tem de ser possível a designação individual de cada uma das cópias.

No que respeita aos sistemas de designação, este objecto pode ser visto como um par que agrupa o seu nome e a lista das suas cópias. A este tipo de par chama-se um grupo.

As vantagens da utilização de grupos são inumeras, saíndo no entanto do espectro deste trabalho. Fica no entanto a referencia sobre a importancia deste conceito.

## DNS-DOMAIN NAME SERVICE

---

### Introdução

O *Domain Name Service* (DNS) é um serviço e designação desenvolvido por Mockapetris em 1987 com o objectivo de substituir o esquema de designação até então utilizado na *Internet*, em que os nomes e os endereços estavam contidos num único ficheiro que era copiado para todos os computadores que dele necessitassem. Este sistema sofria de problemas graves pois por ser centralizado não era extensível a um número elevado de computadores, não permitia a administração local dos nomes e apenas servia para designar endereços de computadores.

Assim, o objectivo era desenhar um sistema em que:

- o espaço de designação fosse consistente sem obrigatoriedade dos nomes conterem identificadores de rede, endereços, etc., como parte deles;
- o espaço de designação fosse distribuído;
- a utilização por vários protocolos fosse permitida;
- a designação de vários tipos de recursos fosse permitida;
- o número de entidades designadas fosse muito elevado.

Para além dos objectivos a cumprir, foram também identificadas algumas características específicas ao sistema onde o DNS iria ser implementado e que influenciam o seu projecto, das quais se destacam:

- baixa velocidade com que os dados variam;
- fronteiras administrativas correspondem a organizações;
- acesso à informação é mais crítico do que a consistência da mesma.

O DNS tem três componentes: espaço de designação, servidores de nomes e *resolvers* que correspondem a três camadas do sistema. Assim temos:

- do ponto de vista do utilizador, o domínio consiste numa árvore única que pode ser acesada através de um simples procedimento para um *resolver* local;
- do ponto de vista de um *resolver*, o domínio é composto por um indeterminado número de servidores de nomes, contendo cada um uma ou mais peças da árvore total.
- do ponto de vista do servidor de nomes, o domínio consiste em conjuntos separados de informação local chamadas zonas. O servidor contém cópias locais de algumas zonas.

Segue-se uma descrição destes três componentes do DNS.

### Espaço de Designação

O espaço de designação do DNS é uma estrutura hierárquica em forma de árvore. Cada nó contém um rótulo que deve ser diferente dos rótulos dos nós irmãos. O nome de um nó arbitrário é a colecção de rótulos no caminho entre ele e a raiz.

O DNS define um domínio como sendo a parte do espaço de designação que está em, ou abaixo do nome que designa o domínio.

A figura 2 representa parte do espaço de designação em uso na *Internet*.

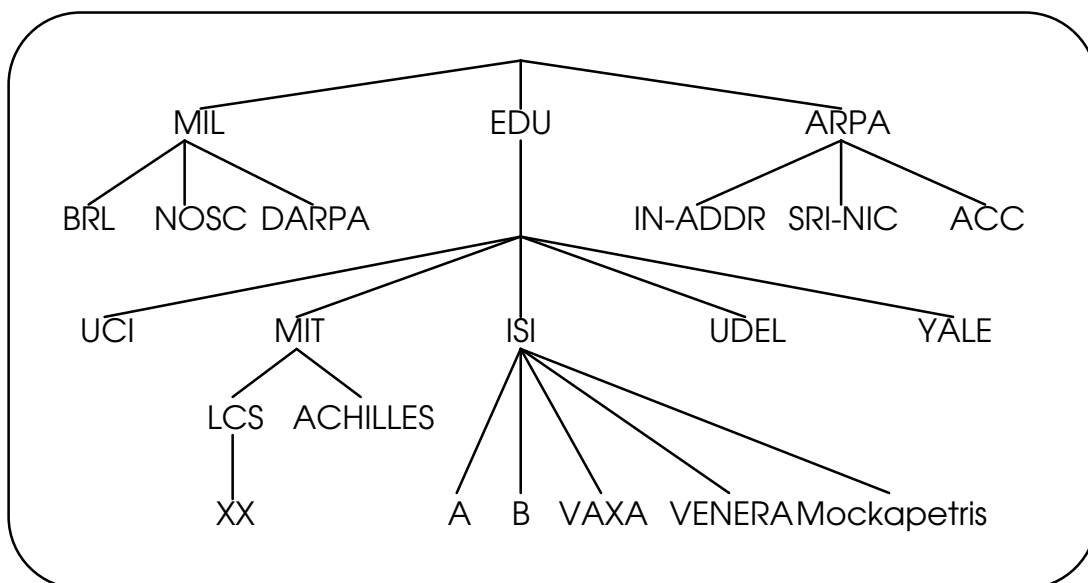


fig.2 - DNS: espaço de designação

Convém referir que apesar de na *Internet* o espaço de designação estar particionado de acordo com critérios organizacionais e geográficos, um objecto cujo nome contenha por exemplo um prefixo referente a uma empresa portuguesa pode estar localizado numa sua sucursal em Espanha, ou seja, o nome de um objecto não o obriga a estar localizado num determinado local.

A cada nó na árvore (nome), corresponde um conjunto de informação de recursos, composto por vários *Resource Records* (RR). Cada um destes RR contém informação de um dado tipo, que pode ser um endereço, um *alias*, um servidor de nomes para o domínio, informação sobre um dado sistema, entre outros.

## Servidores de Nomes

O problema da implementação do DNS em larga escala é tratado através da partição da base de dados bem como da utilização de técnicas de replicação e *caching*.

A base de dados é distribuída por vários servidores. Cada servidor, para além de conter parte da base de dados, contém também referências sobre outros servidores que administrem quer outros domínios, quer sub-domínios administrados separadamente, para que todas as consultas possam ser satisfeitas quer directamente quer através do encaminhamento do pedido para outro servidor.

A base de dados está dividida através de 'cortes' feitos entre nós no espaço de designação. Estes 'cortes' são normalmente feitos em pontos onde outra organização pretende tomar para si a tarefa de administrar o espaço de designação. Depois de todos os cortes feitos, cada grupo ligado do espaço de designação é aquilo a que se chama uma zona.

Assim, cada zona contém:

- atributos para os nomes nesse domínio, menos possíveis sub-domínios pertencentes a outras zonas e administrados por outras autoridades;
- nomes e endereços de servidores de nomes que facultam dados autoritários para essa zona i.e. servidores aos quais os dados foram atribuídos de uma forma 'oficial' e que podem ser confiados como estando razoavelmente actualizados;
- nomes de servidores de nomes que contêm dados autoritários para sub-zonas delegadas, e respectivos endereços.

Para que os dados de uma determinada zona estejam disponíveis se um servidor avariar, o DNS impõe que cada zona seja replicada de uma forma autoritária em pelo menos dois servidores independentes. Um servidor pode conter dados autoritários para zero ou mais zonas.

Existem dois tipos de servidores que contêm dados autoritários para uma zona: um deles é o primário (*master*) que lê a informação directamente de um ficheiro editado pelo administrador; todos os outros são secundários pois carregam periodicamente a sua informação a partir do servidor primário. Esta política de replicação implica que todas as alterações tenham de ser feitas de uma forma centralizada no servidor primário.

Convém referir que todos os servidores podem conter dados de uma forma não autoritária, tendo no entanto que avisar os clientes deste facto cada vez que a resolução de um nome é feita com base neste tipo de informação. Por forma a não fornecer dados significativamente desactualizados, cada entrada numa zona tem um tempo de expiração. Assim, quando um servidor contém dados não autoritários, apenas os poderá utilizar, para resolver um pedido, dentro dos limites impostos por esse tempo de expiração. Este método reduz significativamente o tráfego na rede, preservando uma certa flexibilidade pois o tempo de expiração pode ser atribuído de acordo com a expectativa de vida de cada dado, e garantindo a convergência para um estado consistente a longo prazo (dados incorrectos só serão utilizados durante um certo tempo).

A figura 3 mostra o espaço de designação da figura 2, supondo que se queria um controle administrativo separado para: *raiz*, *MIL*, *EDU*, *MIT.EDU* e *ISI.EDU*. Entre parentesis encontram-se os servidores que gerem as zonas de uma forma autoritária, no ponto da árvore onde eles assumem o controle.

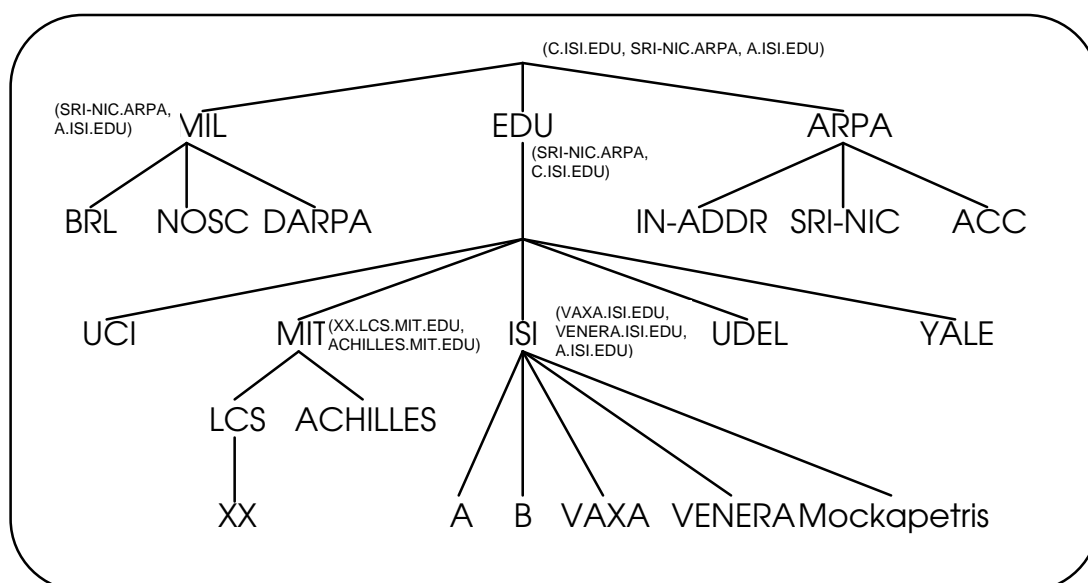


fig.3 - DNS: divisão do espaço de designação

## Resolvers

No contexto do DNS, o *User Agent* é chamado *Resolver*.

Normalmente implementados sob a forma de bibliotecas de *software*, os *resolvers* extraem informação de servidores de nomes em resposta a pedidos dos clientes.

Um resolver tem de conseguir aceder a pelo menos um servidor de nomes e usar a informação fornecida para responder ao pedido ou prosseguir a pesquisa usando as referências para outros servidores. A arquitectura do DNS permite os métodos de navegação iterativo e recursivo, sendo apenas o primeiro obrigatório. O tipo de navegação utilizado é especificado pelo *resolver*.

Por forma a aumentar a eficiência, os *resolvers* mantêm *caches*, utilizando a mesma política dos servidores de nomes no que respeita ao tempo de expiração de uma dada entrada.

## Comentários

Na sua implementação actual na *Internet*, o DNS obtém bons resultados no que se refere ao tempo de resposta, dada a dimensão do sistema em causa. Estes resultados são obtidos à custa de uma boa combinação de partição, replicação e *caching* dos dados.

No que se refere ao tipo de objectos que o DNS permite designar, embora na sua actual implementação ele apenas seja utilizado para mapeamentos entre computadores e endereços, ele pode ser alargado a outro tipo de recursos através da utilização de *Resource Records* de vários tipos por nome.

O DNS impõe a limitação de um recurso de um dado tipo por nome, ou seja, não é extensível a situações onde existam várias instancias de um dado tipo de objecto como por exemplo na designação de ficheiros, nem permite a implementação de grupos. Refira-se no entanto que o DNS foi desenhado para coexistir com outros sistemas de designação locais que mantêm dados mais pertinentes a um nível local como por exemplo o NIS da *Sun* que guarda *passwords* codificadas.

Quanto à questão da consistência dos dados, o DNS permite um certo grau de inconsistência introduzido quer pelo desfazamento entre as actualizações no primário e no secundário, quer pelo uso de *caches*. Estes estados inconsistentes não são graves devido à lenta velocidade com que os dados variam. De qualquer forma, é garantida a convergência para um estado consistente.

As duas grandes desvantagens do DNS são por um lado a forma centralizada que é imposta para as actualizações, tendo estas de obrigatoriamente feitas no servidor primário através do administrador, e, por outro a sua rigidez no que respeita a alterações na estrutura do espaço de designação, não permitindo em termos práticos, por exemplo, mover um domínio para debaixo de outro domínio. Como será descrito no próximo capítulo, o *Global Name Service* aborda e resolve estas duas questões.

## GNS-GLOBAL NAME SERVICE

---

### Introdução

Desenvolvido por Lampson e apresentado em 1986, o *Global Name Service* (GNS) é um sistema de designação baseado nos sistemas *Grapevine* e *Clearinghouse* com o objectivo de fornecer meios para localização de recursos, edereçamento de correio e autenticação num sistema distribuído.

Lampson começa por identificar a grande diferença entre um sistema de designação distribuído e uma base de dados genérica distribuída. Esta diferença reside particularmente em dois pontos:

- num sistema distribuído, os nomes e seus atributos variam muito lentamente pelo que, ao contrário de uma base de dados genérica, um sistema de designação pode ser concebido tendo em atenção este ponto, que torna o seu desenho bastante mais simplificado.
- as restrições no que respeita à integridade de um sistema de designação são muito mais 'leves' do que as de uma base de dados genérica, nomeadamente no que respeita às exigências de consistência.

São também identificadas as diferenças entre um sistema de designação distribuído e um sistema de ficheiros no que respeita à velocidade necessária para criar e consultar nomes (muito maior no sistema de ficheiros), e ao tamanho e disponibilidade (muito menor no sistema de ficheiros).

Os objectivos que o GNS se propõe alcançar são idênticos aos já referidos no primeiro capítulo. Recapitulando temos:

- tamanho arbitrariamente elevado;
- tempo de vida elevado suportando reestruturações;
- disponibilidade elevada;
- isolamento de falhas;
- tolerância a entidades suspeitas.

A solução adoptada para alcançar estes objectivos passa por uma estrutura hierárquica do sistema, por este ser o método fundamental para acomodar alterações e isolar falhas.

Segue-se uma descrição do sistema sob dois níveis distintos:

- ao nível do cliente, existem nomes hierárquicos e seus valores, operações para os consultar e alterar bem como facilidades de protecção e autenticação. Neste nível o sistema é transparente no que toca à sua distribuição.
- ao nível do administrador, as várias cópias replicadas da base de dados são visíveis bem como os mecanismos de localização e sincronização dessas mesmas cópias.

Por último, será feita uma descrição mais pormenorizada dos mecanismos utilizados para lidar com a alteração na estrutura do espaço de designação.

### Cliente

Do ponto de vista de um cliente do GNS, a estrutura do sistema é muito semelhante àquela apresentada pelo sistema de ficheiros do *UNIX*.

Esta estrutura é composta por uma árvore de directorias (ver fig.4), onde cada uma é identificada por um inteiro que representa o identificador de directoria (DI) único e por um nome pelo qual pode ser acedida pelo pai. Os arcos da árvore são chamados referências de directoria (DR) e

são os valores dos nomes. Os DR's são os DI's das directorias filhas. Assim, uma directoria pode ser designada relativamente a uma raiz, pelo seu caminho, que é chamado o nome completo (FN) da directoria.

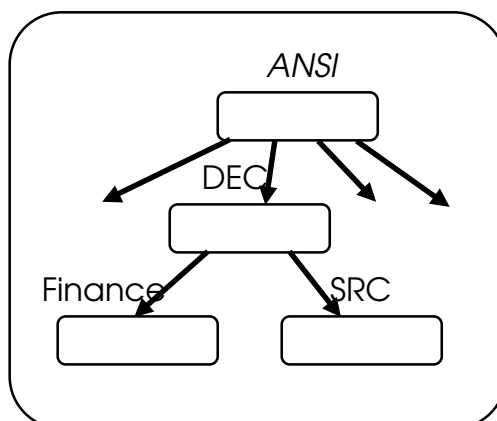


fig.4 - GNS: árvore de directorias

No exemplo da figura 4, a directoria inferior é designada DEC/SRC relativamente à raiz rotulada ANSI. Assim o seu FN será ANSI//DEC/SRC onde o *ANSI* representa o DI da raiz.

O valor de um nome pode ser um *link*. Neste caso, esse valor será o FN de uma outra directoria.

A grande diferença entre o GNS e outros modelos de designação reside no facto de cada directoria não ser simplesmente um mapeamento de nomes em valores. Assim, cada directoria contém uma árvore de valores (ver fig.5).

Cada arco da árvore de valores contém um rótulo (L), e cada nó uma estampilha temporal (TS) bem como uma indicação sobre o seu estado de presença/ausencia. Um caminho é definido por uma sequencia de rótulos (L\*), escrita na mesma forma que um FN.

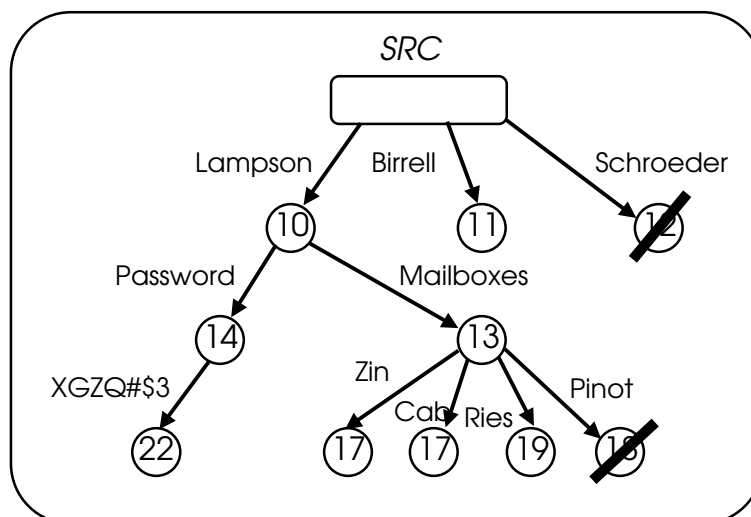


fig.5 - GNS: árvore de valores

Quanto ao valor de um determinado caminho, podemos distinguir três casos distintos:

- se o caminho  $L^*/l_i$  termina numa folha que é filha única, diz-se que  $l_i$  é o valor de  $L^*$  (ex: para o caminho Lampson/Password/XGZQ#\$3, diz-se que XGZQ#\$3 é o valor de Lampson/Password).
- se o caminho  $L^*/l_i$  termina numa folha que não é filha unica, e os seus irmãos têm rótulos  $l_1...l_m$ , diz-se que o conjunto  $\{l_1...l_m\}$  é o valor de  $L^*$  (ex: {Zin, Cab, Ries, Pinot} é o valor de Lampson/Mailboxes).

- se o caminho  $P^*$  não termina numa folha, diz-se que a sub-árvore com raiz no nó onde o caminho termina é o valor de  $P^*$  (ex: o valor de Lampson é a sub-árvore com raiz no nó com TS=10).

Temos assim que um nome é composto por duas partes: o nome completo (FN) de uma directoria, e o nome de uma entidade contida nessa directoria. No contexto das figuras 4 e 5, *ANS//DEC/SRC/Lampson* é o nome de Lampson na directoria *ANS//DEC/SRC*. Este nome tem uma propriedade *Password* com o valor *XGZQ#\$3* e uma propriedade *Mailboxes* com o valor {Zin, Cab, Ries, Pinot}. Um nó arbitrário tem um nome composto por duas partes como por exemplo (*ANS//DEC/SRC, Lampson/Mailboxes*).

No que respeita às operações disponíveis, temos que uma actualização a uma directoria faz com que o nó no fim de um caminho fique presente ou ausente. Adicionalmente todas as actualizações têm a si associadas um TS com o objectivo de permitir actualizações concorrentes de locais diferentes sem sincronização prévia como será descrito adiante. Fica só como referencia o facto de um TS mais recente ter precedencia sobre um TS mais antigo.

Quanto á autenticação, não será tratada neste trabalho por sair do espectro proposto para o mesmo. Fica apenas a noção de que o controle de acesso é feito com base na noção de entidades (*principal*) que podem ser autenticadas mediante o seu conhecimento de uma chave de encriptação.

## Administrador

O administrador tem por função a atribuição de recursos para a implementação do serviço, e a sua reconfiguração para lidar com falhas.

Em vez de uma directoria, o administrador vê um conjunto de cópias de directoria (DC), residentes em servidores distintos (ver fig.6). Assim, uma referencia de directoria (DR) inclui para além do DI da directoria, uma lista de servidores que guardam as DC's.

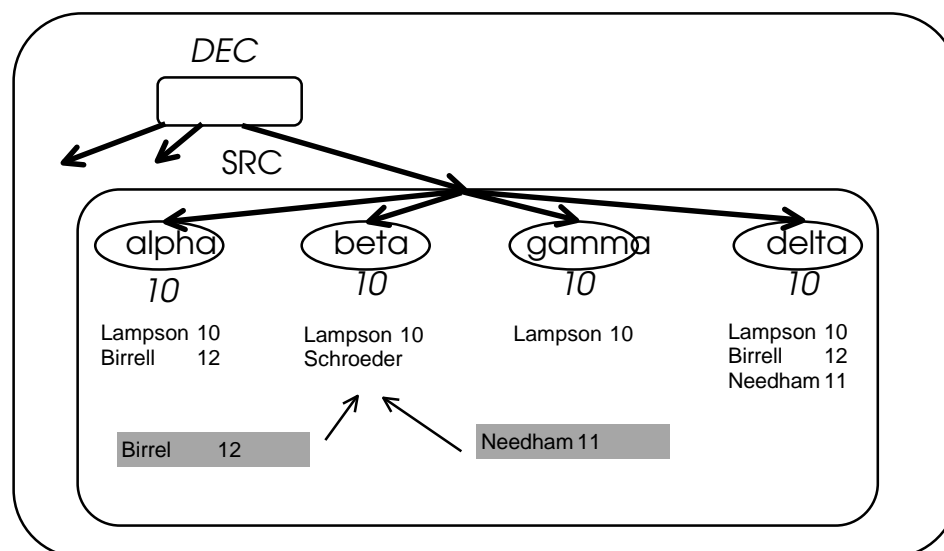


fig.6 - GNS: cópias de directoria com conteúdos diferentes

As várias DC's não são exactamente iguais. Isto deve-se ao facto, como já foi referido anteriormente, de o GNS permitir aos clientes a realização de actualizações concorrentes, nas diversas DC's. O GNS utiliza um método que embora não garanta a completa consistencia entre as várias DC's, garante a convergencia das mesmas para um estado consistente. Assim, a cada actualização feita numa DC, é associada uma estampilha temporal (TS). Podemos ver na figura 6 as várias DC's com actualizações efectuadas com TS's de 10,11,12 e 14. As várias cópias estão consistentes no que respeita a actualizações efectuadas com TS até 10 (número indicado em itálico), que é o chamado tempo do último varrimento (*sweep*).

Existem dois métodos pelos quais se podem espalhar as actualizações pelas várias cópias:

- o primeiro consiste no envio de mensagens entre as várias DC's, contendo as actualizações (ver fig.6);
- o segundo, chamado varrimento (*sweep*), consiste na visita a todas as DC's para recolher as actualizações efectuadas em cada uma delas, e posterior actualização de todas as DC's tendo em conta o estado final produzido pelo conjunto de todas entradas recolhidas.

A periodicidade do varrimento é determinada pelo administrador e, a cada um, está associada uma estampilha temporal. A figura 7 mostra a directoria da figura 6 após um varrimento com estampilha temporal de 14.

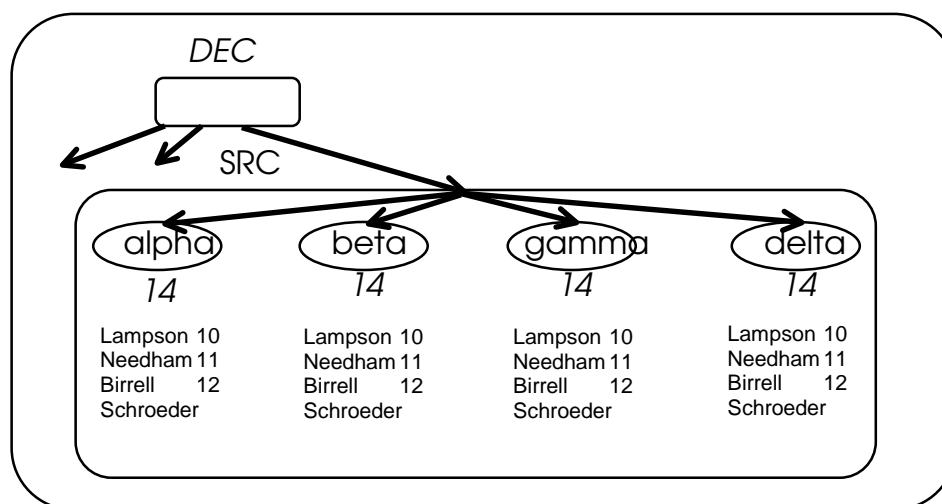


fig.7 - GNS: a directoria da fig. 6 após um varrimento

O primeiro método referido, o das mensagens, é sem dúvida o mais rápido. No entanto, a dificuldade na sua implementação com total eficácia torna necessário recorrer aos varrimentos periódicos, mais simples de realizar de uma forma eficaz.

Por forma a garantir a recolha das actualizações de todas as DC's sem ter que recorrer ao DR na directoria ascendente pois esta pode estar a ser alterada durante o varrimento, os servidores que as gerem estão organizados segundo um anel lógico.

No caso de avaria de um servidor contido no anel, cabe ao administrador a criação de uma nova sequencia fechada de servidores que, daí para a frente passarão a gerir as DC's, sem no entanto evitar a perda das actualizações efectuadas na DC contida no servidor avariado.

## Espaço de Designação

Nesta secção serão descritos os métodos utilizados para permitir o crescimento e a reestruturação do espaço de designação. Serão também descritos os métodos utilizados para o uso de *caches*.

O mecanismo básico para o crescimento do espaço de designação é a sua estrutura hierarquica que permite a criação de nomes numa directoria de uma forma completamente independente daquilo que se passa nas outras directorias, bem como a criação de outras directorias de uma forma extremamente simples.

No que respeita ao crescimento por combinação de serviços de designação distintos, previamente existentes, o problema é um pouco mais complicado, sendo resolvido do seguinte modo: cria-se uma nova raiz, e fazem-se as raízes das árvores a combinar descendentes desta nova raiz (ver fig.8). Para resolver o problema criado por esta solução no que respeita à resolução dos nomes antigos no novo espaço de designação, como por ex. na fig.8 o nome DEC/SRC/Lampson passa a ser ANSI/DEC/SRC/Lampson (na figura, os nomes aparecem com

o DI explicitamente indicado, começando por #), é mantida na nova raiz uma tabela de mapeamentos entre alguns DI's (*Well Known Dirs*) em *links* que são FN's relativamente à raiz (ver fig.8). Assim, quando uma consulta chegar à raiz, pode substituir o DI do FN com o caminho que começa nesa mesma raiz. De referir que ainda que existem mecanismos para permitir que todos os servidores possam aceder a uma cópia da raiz.

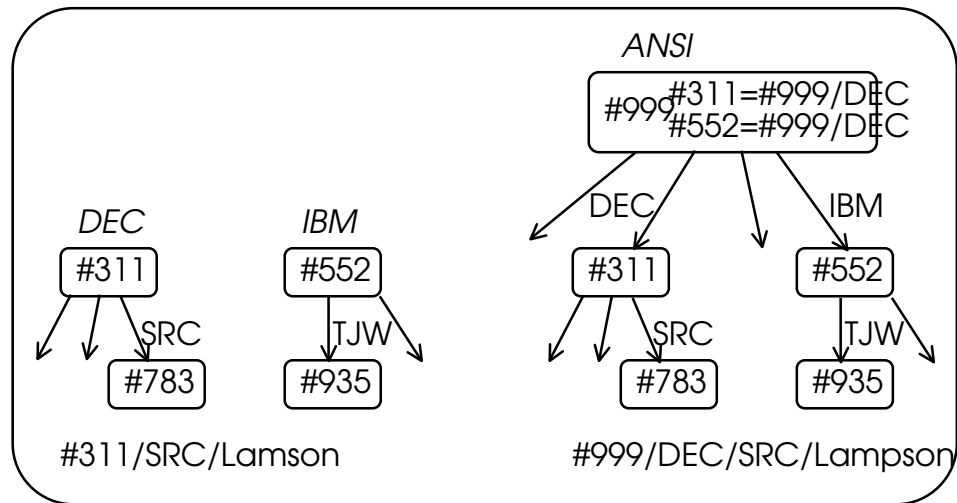


fig.8 - GNS: combinação de duas árvores sob uma nova raiz

Quanto à reestruturação do espaço de designação, vamos supor por exemplo que a DEC compra a IBM. A sub-árvore com raiz na directoria IBM deveria ser transferida para debaixo da directoria DEC (ver fig.9). Para resolver este caso, é criado um novo DR na directoria DEC que referencia a directoria IBM e a entrada de IBM na directoria ANSI passa a conter um link para o novo local (ver fig.9), desde que não sejam introduzidos ciclos no espaço de designação.

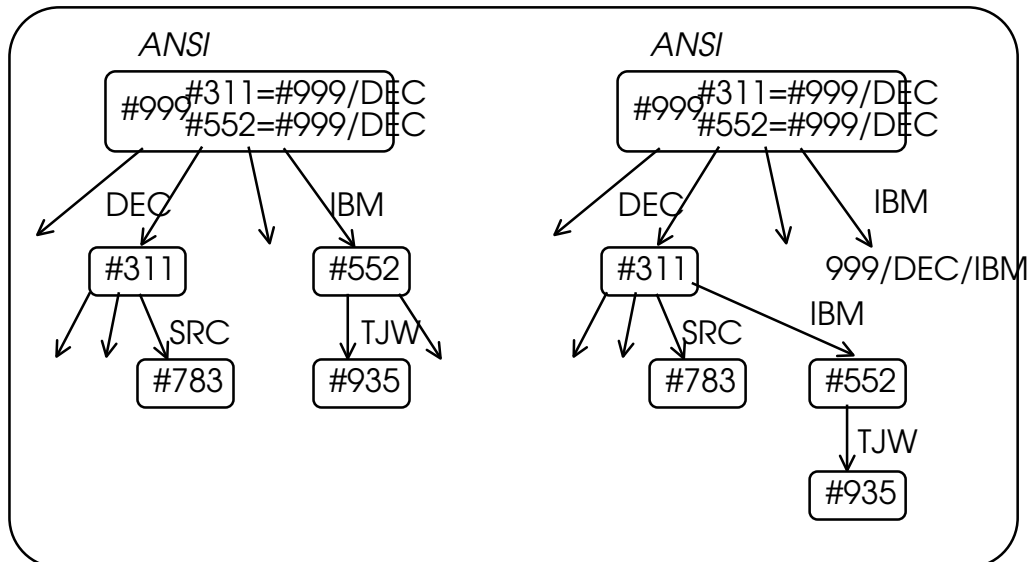


fig.9 - GNS: reestruturação do espaço de designação

A utilização de *cache* para guardar as consultas ao serviço de designação é fundamental para uma boa eficiência do sistema em termos de custo e de velocidade. Devido ao facto de ser impraticável notificar um cliente sobre uma alteração a algum valor por ele mantido em *cache*, a utilização deste tipo de memória tem de ser paga de uma das seguintes formas:

- permitir algumas inconsistências nos valores mantidos em *cache*, solução que não necessita de nenhum mecanismo especial;
- forçar a lenta variação dos dados.

O GNS introduz um mecanismo particular para forçar a lenta variação dos nomes no espaço de designação e conseqüente aumento da validade dos valores mantidos em *cache*. Assim, para qualquer entrada no espaço de designação e em particular para os arcos na árvore de directorias, é associado um tempo de expiração (TX) durante o qual essas entradas não podem ser alteradas. Com este mecanismo garante-se a validade de uma entrada na *cache* até ao menor dos valores de TX encontrados. Na figura 10, o resultado da consulta de ANSI/DEC/SRC é válido até 15/SET/1994.

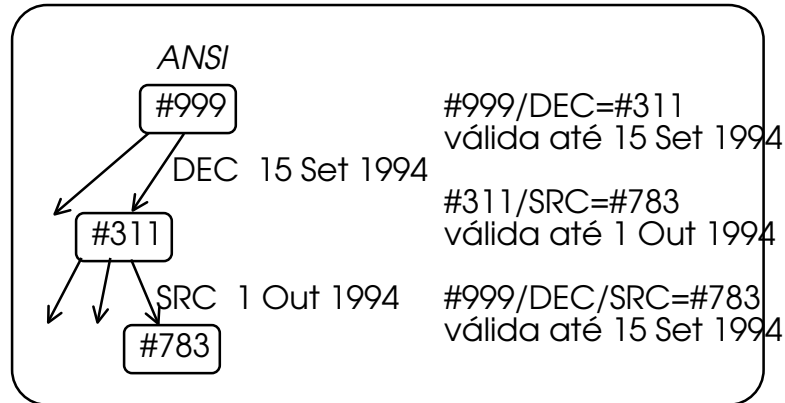


fig.10 - GNS: utilização do tempo de expiração para validar as *caches*

## Comentários

O *Global Name Service* é um sistema que tem como referencias positivas a forma como resolve o problema da inflexibilidade do espaço de designação no que se refere á sua reestruturação, e o problema da centralização das actualizações, ambos apresentados pelo DNS.

A solução proposta para a descentralização das actualizações, embora introduzindo algum grau de inconsistencia é completamente aceitavel dada a natureza e as necessidades de um sistema de designação.

Quanto à reestruturação do espaço, tem um unico senão que está relacionado com as necessárias tabelas de '*Well Known Dirs*' que introduzem um ponto de centralização que pode crescer demasiado com o tempo. Na prática, e dada a situação actual no que repeita à organização do espaço, uma eventual implementação em larga escala do GNS não teria que suportar muitas alterações relacionadas com a junção de duas árvores distintas donde este problema não seria significativo.

Outra grande vantagem do GNS em relação ao DNS é a sua potencialidade para designar um espectro muito alargado de objectos, e muito particularmente a capacidade para implementar grupos (ex. da fig.5 onde o valor de Lampson/Mailboxes é {Zin, Cab, Ries, Pinot}).

## X500-DIRECTORY SERVICE

### Introdução

Deenvolvido pela ISO e CCITT, e integrado no conjunto de standards OSI (*Open Systems Interconnection*), o X.500, também conhecido por 'A Directoria', tem por objectivo descrever um serviço que, para além da habiul conversão entre nomes e endereços, permita a designação de todo e qualquer objecto do mundo real.

Utilizndo uma designação baseada em atributos, o X.500 pretende por um lado possibilitar aos individuos e organizações a diponibilização de um largo espectro de informação sobre eles próprios e os recursos que pretendem oferecer, e por outro, permitir aos utilizadores a pesquisa de informação especifica sem total conhecimento do nome, estrutura ou conteúdo.

Segue-se uma descrição do modelo da informação e do modelo funcional do X.500.

### O Modelo de Informação

A informação está contida na 'Base de Informação da Directoria' (DIB) sob a forma de entradas. Assim, cada um destes elementos contém uma colecção de atributos referentes ao objecto que designam, de uma forma muito semelhante a um *record* numa base de dados convencional. Para cada entrada existe um atributo referente ao objecto que ela contém, que define quais os atributos obrigat'rios e facultativos que devem figurar. As entradas são classificadas numa forma em tudo similar às estruturas de classes de objectos encontradas nas linguagens orientadas para este tipo de programação. Assim, a definição da classe de um objecto é baseada no conceito de herança de classe, querendo isto dizer que pode ser definida como sendo uma sub-classe de uma classe préviamente definida, com refinamentos adicionais para além de toda a estrutura herdada.

No que respeita à estruturação da DIB, as entradas são designadas de uma forma única usando uma estrutura hierárquica chamada 'Árvore de Informação da Directoria' (DIT). O nome de uma entrada, chamado o 'Nome Distinto' (DN), é formado pelo DN da entrada ascendente ao qual é acrescentado o seu 'Nome Distinto Relativo' (RDN) . Cada entrada contém um ou mais valores de atributos chamados 'distintos' que em conjunto formam o RDN dessa entrada, e que a devem distinguir de todas as suas irmãs na DIT. Assim, o DN é o conjunto de todos os RDN's encontrados desde a raiz até à entrada em questão.

Na figura 11 podemos ver um pequeno exemplo de uma DIT com referencia para o RDN e DN de um dos ramos da árvore. Convém referir que na figura apenas se representam os atributos que contribuem para o RDN de cada entrada, podendo existir mais.

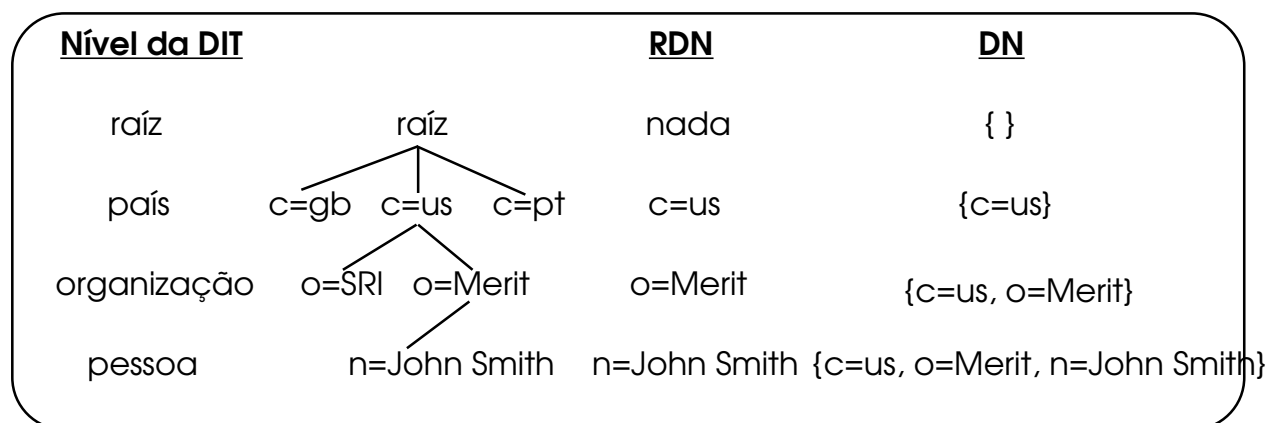


fig.11 - X.500: *Directory Information Tree*

Por fim convém referir que este modelo permite o uso de *alias*.

## O Modelo Funcional

O cliente da directoria acede-a através de um 'Directory User Agent' (DUA) estabelecendo com ele uma relação de 1:1. A composição interna da Directoria é completamente invisível ao cliente. Como veremos mais adiante, o próprio DUA também não necessita saber muito sobre essa composição interna.

O modelo funcional descreve as interações dentro da Directoria. Assim, o DIB é composto por fragmentos que estão contidos em 'Directory System Agents' (DSA). Estes DSA's cooperam entre si para fornecer o serviço da Directoria.

Quando um DUA contacta com um DSA, duas situações podem ocorrer: o DSA contém a informação necessária ou não. No primeiro caso, a informação é simplesmente enviada para o DUA (ver fig 12).

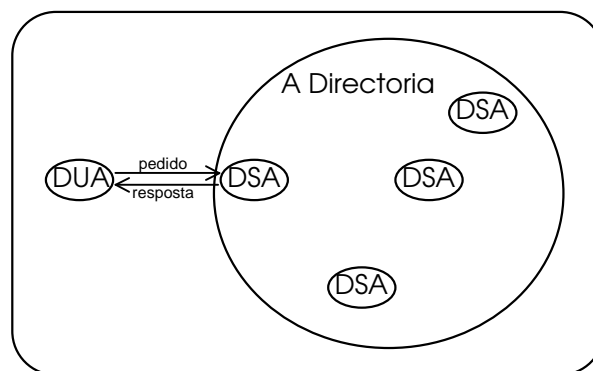


fig.12 - X.500: pedido/resposta

No segundo caso, o DSA tem de ser capaz de identificar outro DSA que esteja de alguma forma 'mais perto' da informação e optar por:

- devolver ao DUA o nome e endereço desse DSA (método iterativo conhecido por *referral* no contexto do X.500). O DUA é então responsável por estabelecer o contacto com o novo DSA (ver fig.13).

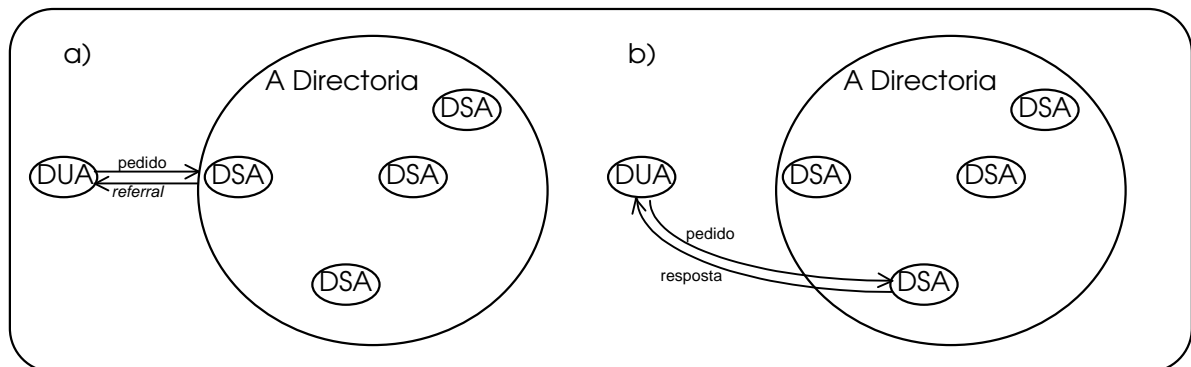
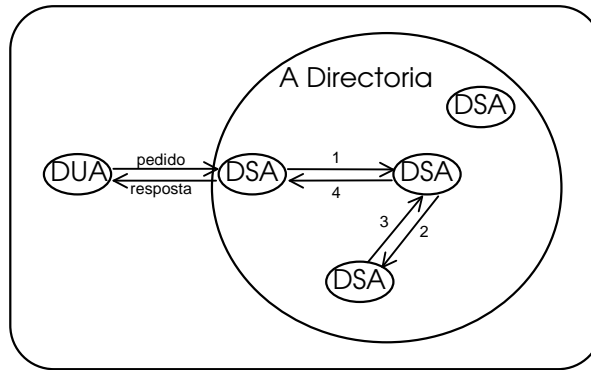
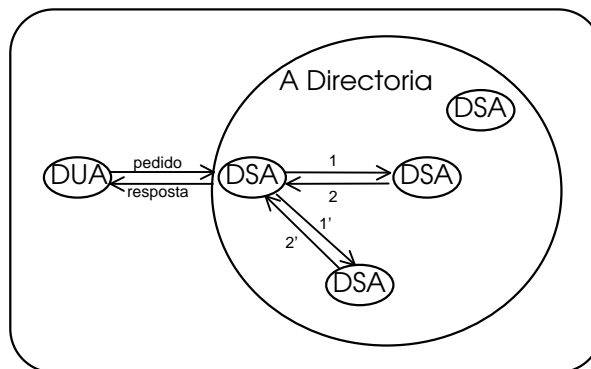


fig.13 - X.500: a)pedido/referral b)pedido/resposta

- contactar directamente com o novo DSA, agindo como um DUA (ver fig.14). Esta técnica recursiva designa-se por *chaining*.

fig.14 - X.500: *chaining*

Se um DSA optar pelo *chaining*, pode ter que contactar diversos DSA's para obter a info desejada, podendo esta operação ser realizada em paralelo utilizando técnicas de multicasting (ver fig 15).

fig.15 - X.500: *chaining em paralelo*

Convém referir que todos estes modos de operação podem ser utilizados em diferentes alturas para resolver um único pedido de um DUA. Um possível critério para a escolha do método seria o 'preço' das ligações estabelecidas.

Resta agora descrever a forma como a DIT está distribuída pelos diversos DSA's, bem como os mecanismos presentes para que um DSA possa saber onde encontrar outro DSA 'mais perto' da informação.

Assim, considera-se a DIT como sendo composta por várias sub-árvores não intersectadas, estando cada uma totalmente contida num DSA. A estas sub-árvores chamamos contextos de designação (ver fig.16).

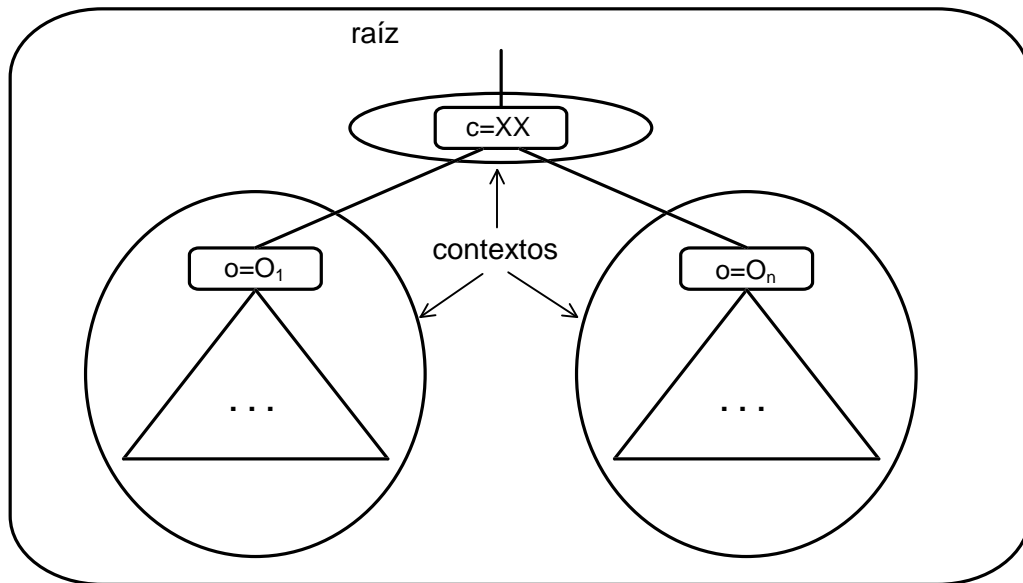


fig.16 - X.500: divisão da DIT em contextos

Introduzindo os conceitos de: 'superior'(relativamente a um contexto), como sendo o contexto que contém as entradas imediatamente acima, e 'subordinado' como sendo o sub-contexto que contém as entradas imediatamente abaixo, temos que para um dado DSA poder resolver um pedido, ele tem que conter:

- pelo menos uma referência para um DSA superior;
- referências para todos os DSA's subordinados;
- informação completa sobre as entidades do seu contexto.

No que respeita à replicação da informação por vários DSA's, o standard definido em 1988 não especifica nada sobre este assunto. No entanto, algumas implementações do X.500 utilizaram técnicas de replicação para aumentar a eficiência do serviço, estando prevista a inclusão deste tópico na nova versão do standard.

## Comentários

Introduzindo uma arquitectura substancialmente diferente dos sistemas anteriormente descritos, o X.500 tem por objectivo ser uma solução global no domínio da designação.

Assim, a sua estruturação utilizando as noções de classes de objectos e em particular a possibilidade de serem criadas classes locais, permitem-lhe o alargamento do seu universo a tudo o que se deseje designar através dele. Isto no entanto já era de certa forma permitido pelo GNS com as suas árvores de valores dentro de uma directoria.

Por outro lado, a designação baseada em atributos permite a realização de pesquisas complexas, nomeadamente em casos onde não se tem total conhecimento do nome a resolver.

No entanto, aquilo que seria a sua grande vantagem, a globalidade, acaba por se revelar o seu grande *handicap*. Ao tentar ser uma solução global, o X.500 introduz potencialidades teóricamente muito interessantes mas de duvidosa aplicação prática. Assim, o X.500 revela-se demasiado 'pesado', perdendo eficiência nas tarefas mais comuns requeridas a um sistema de designação.

Outro dos grandes problemas do X.500 está relacionado com a incompletude do standard apresentado em 1988 que, por exemplo, não incluía a replicação da base de dados, requisito fundamental para uma boa performance num sistema distribuído. Este problema já é abordado no standard de 1992 mas é convicção de alguns autores que o X.500 está condenado à

nascença a não se tornar uma solução universal e única para resolver o problema da designação nos sistemas distribuídos.

## CONCLUSÕES

---

Ao longo destes capítulos foram abordados vários tópicos relacionados com a designação em sistemas distribuídos, apontando as suas necessidades e descrevendo algumas soluções adoptadas.

Foram apresentados três soluções distintas com características bastante diferentes.

O *Domain Name System*, largamente utilizado na *Internet* para designação de computadores, obtém uma boa resposta sendo no entanto algo limitado no que se refere à descentralização das actualizações, reestruturação do espaço de designação e flexibilidade no que respeita aos objectos que permite designar. Todos estes problemas foram particularmente abordados e resolvidos pelo *Global Name Service*. Por último o X.500 apresenta a mais ambiciosa das soluções com a sua designação baseada em atributos, sendo que do ponto de vista prático talvez peque por excesso de generalidade.

Existem no entanto alguns pontos comuns a todos os sistemas de designação distribuídos que ambicionem uma boa performance. Eles residem numa boa partição do espaço de designação e no uso eficiente de técnicas de replicação e *caching*.

**BIBLIOGRAFIA**

---

- Cheriton D.R. and Mann T.P. (1986). *A Decentralized Naming Facility*. Computer Science Dep.- Stanford University
- Terry D. B. (1987). *Caching Hints in Distributed Systems*. IEEE Trans. Software Eng. vol SE-13 no 1
- Lampson B. W. (1986). *Designing a Global Name Service*. Proc. 5th Annu. ACM Symp. Principles of Distributed Computing, pp. 1-10
- Needham, R. (1989). *Names*. In Distributed Systems, an Advanced Course. (Ed. Mullender, S.), First Edition, Workingham: ACM Press/Addison-Wesley. PP. 315-26
- Rose, Marshall T. (1992). *The Little Black Book: Mail Bonding with OSI Directory Services*. Prentice.Hall, Englewood Cliffs NJ
- Goscinski, A. (1991). *Naming Facility*. In Distributed Operating Systems, the Logical Design. Addison-Wesley. PP. 285-368
- Watson, R. (1985). *Identifiers (naming) in distributed systems*. In Distributed Systems,- Architecture and Implementation - an Advanced Course. (Ed. Lampson, B.), Springer-Verlag PP. 191-210
- Legatheaux, M. (1988). *La désignation dans les systèmes d'exploitation répartis*. Technique et Science Informatiques vol 7, n° 4
- Mockapetris, P.(1987). *Domain Names - Concepts and Facilities*. Tech Report RFC 1034
- Weider, C.(1992). *Technical Overview of Directory Services Using the X.500 Protocol*. Tech Report RFC 1309