Federal de Pernambuco
Centro de Informática

UM ESTUDO DA INTEGRAÇÃO DO MICROSOFT ACTIVE DIRECTORY COM FONTES DE DADOS DIVERSAS

Trabalho de Graduação

Aluno: Sérgio Silveira Clemente Filho

Orientador: André Luís de M. Santos

Recife, 09 de Fevereiro de 2006

One thing is sure. We have to do something. We have to do the best we know how at the moment... If it doesn't turn out right, we can modify it as we go along.

Franklin D. Roosevelt

Agradecimentos

Quero agradecer a todos que me apoiaram e ajudaram durante toda minha graduação, sem os quais eu não poderia ter realizado este trabalho:

- Ao meu amor Isabelle, por todo apoio e incentivo que me deu;
- Aos meus pais e familiares, que sempre me incentivaram durante a minha jornada;
- A todos meus amigos do Centro de Informática, em especial: Vaco, Dudu, Pacheco, Guto, Chico, Boró, Dap, Mano, Kval, Osandy e muitos outros que não constam nesta lista;
- Aos meus companheiros de trabalho, especialmente aos meus chefes Fábio Ávila e Romulo Martins por terem gerenciado o projeto UniProv que serviu como base para este trabalho e por terem financiado muitas cachaças;
- Ao meu orientador André Santos por me guiar durante a elaboração deste trabalho;
- A todos os professores do Centro de Informática, que contribuíram para minha formação profissional;
- A todos amigos em geral, especialmente a aqueles que me chamavam varias vezes pra tomar uma e eu farrapei várias vezes para ficar elaborando este trabalho.

Resumo

O servidor de diretórios Microsoft Active Directory (MS-AD) contém informações sobre usuários, recursos da rede e controla a autenticação da rede Windows. A manipulação de usuários do Windows normalmente é feita de forma manual e nem sempre tem relação com o sistema de recursos humanos.

Além disso, a existência de ambientes mistos é um fato comum nas corporações. Não é raro encontrar concomitantemente sistemas operacionais Linux e Windows, bancos de dados Microsoft Sql Server e Oracle. Cada sistema desses possui seu repositório de usuários e utiliza uma forma de autenticação peculiar.

Este trabalho se propõe a estudar as tecnologias disponíveis para o gerenciamento automático do MS-AD através de fontes de informação diversas. Além disso, investigar a sincronização do MS-AD com outros servidores de diretórios.

Palavras chave: Microsoft Active Directory, servidor de diretórios, integração, fonte de dados diversas.

Sumário

Índice de Figuras

Índice de Tabelas

1 Introdução

Atualmente, num ambiente corporativo, é bastante comum a existência de um grande número de aplicativos com propósitos distintos. Podem ser aplicativos simples de intranet como, por exemplo, para a reserva de salas, controle de impressão, etc; ou um pouco mais elaborados como um sistema de recursos humanos, aplicativos de colaboração como o Microsoft Sharepoint [MS Sharepoint], etc.

Cada aplicação possui seu repositório de usuários, realiza a autenticação[1] dos mesmos de alguma forma e realiza do controle de acesso[2]. Vários métodos de autenticação podem ser empregados, como reconhecimento de impressão digital, voz, assinatura, íris, etc. Porém o reconhecimento através do login e senha ainda é o mais utilizado hoje em dia devido a sua simplicidade.

Cada aplicação tem sua definição própria de usuário (nome, título, login, senha, cargo e grupos a qual pertence) e normalmente emprega uma forma de autenticação peculiar. Além disso, cada uma normalmente possui sua própria ferramenta para gerenciamento de usuários, grupos, políticas, etc. A Figura 1 é um exemplo clássico da redundância existente.

[1] É o ato de um computador/usuário provar que é ele mesmo.

[2] É o ato de restringir o acesso somente a pessoas autorizadas. Por exemplo, uma recepcionista de um sistema médico não vai ter acesso à parte financeira do sistema.

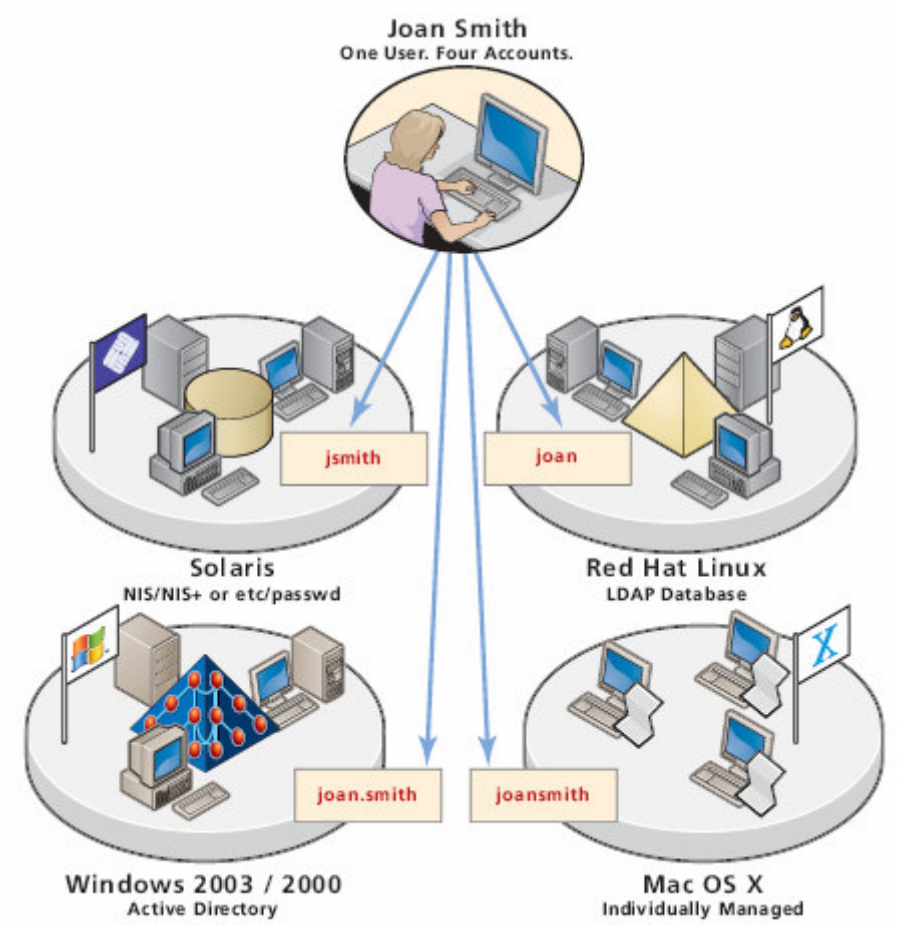

Figura 1 - Redundância de contas de usuários. Fonte: [Centrify2]

A usuária final Joan Smith possui quatro contas de usuários, para os ambientes Windows, Solaris, Red Hat Linux e Mac OS. Podemos observar claramente a conta de um mesmo usuário replicado em vários repositórios. Há uma sobrecarga no usuário final que se precisa lembrar da senha em vários ambientes.

Segundo [Centrify] 45% das chamadas ao helpdesk são para resetar a senha, aumentando os custos da organização de TI na medida em que o número de repositórios de usuários cresce. Segundo essa mesma fonte, uma grande organização tem mais de vinte repositórios de usuários, resultando numa média de mais de cinco pares de logins e senhas por usuário final.

Conforme mencionado em [Centrify2] “In a typical IT environment, heterogeneity is the standard”, ou seja, é comum encontrar uma organização comcomitantemente com sistemas operacionais Unix e suas variantes, OS/2, NetWare e Windows, bancos de dados Oracle, DB2, SQL Server ao mesmo tempo. Cada sistema desses

normalmente possui seu repositório de usuários e utiliza uma forma de autenticação particular.

Problemas típicos desse tipo de ambiente são:

- Esses repositórios de usuários são independentes e não possuem nenhuma integração entre si.
- Administrar esses múltiplos repositórios normalmente leva a um grande desperdício de tempo além do tremendo esforço para manter essas bases atualizadas. Pois, não existe uma ferramenta única para atualizar todos os repositórios.

Além dos problemas supracitados, existe outro problema: A interligação desses usuários com o sistema de recursos humanos. Por exemplo, quando uma pessoa é contratada, um responsável do setor de recursos humanos é alocado para fazer o registro no sistema de recursos humanos (SRH). Logo após o registro, o helpdesk precisa criar a conta no Microsoft Active Directory (MS-AD), assim como as contas em todos diretórios e repositórios necessários que o usuário necessitar.

Claramente, isso pode se tornar difícil de controlar e manter a sincronização das informações. Os principais problemas são citados a seguir:

- Novos empregados são criados no sistema de recursos humanos, mas demoram dias para poderem utilizar as aplicações internas.
- Empregados são demitidos e continuam com acesso a programas e recursos por meses ou anos.
- Empregados não possuem seus dados atualizados nos repositórios. Assim se o empregado mudar de e-mail, departamento, telefone, seus antigos dados não serão replicados nos repositórios.

Algumas iniciativas já foram feitas como o .NET Passport [NET Passport], que permite que se utilize uma única credencial para vários aplicativos distintos. O usuário final basta se lembrar do seu e-mail e da sua senha e consegue acesso a todos os aplicativos que utilizarem essa tecnologia para autenticação de seus usuários. O .NET Passport pode ser utilizado nas plataformas Windows e Linux [Net

Passport2], porém além essa alternativa não é gratuita, ela não resolve o problema de controle de acesso, forçar políticas, etc. Vale salientar que a tecnologia é proprietária.

Segundo a pesquisa conduzida pela [TechTarget], 60% das organizações IT já implantaram o MS-AD e 30% planejavam até o final de 2005. O ambiente Linux/Unix precisam lidar com uma vasta gama de repositórios de usuários, são eles:

- Arquivos /etc/passwd em host individuais.
- Grande uso do Sun Microsystems Network Information Service (NIS), que a Sun já avisou que não vai mais prestar suporte.
- Uso de soluções legadas como o Netscape Directory.

Devido a grande presença do MS-AD nas corporações, o MS-AD foi escolhido nesse trabalho como repositório principal de usuários, controlando a autenticação e políticas no ambiente corporativo. Os outros repositórios deverão se integrar com o MS-AD de alguma forma. A Figura 2 ilustra o MS-AD como figura central numa organização.

Figura 2 - Utilização do MS-AD num ambiente heterogêneo. Fonte: [Centrify]

Seguindo essa tendência, a Oracle, através da Oracle Directory Integration Platform [OracleIntegrationPlatform] permite que os usuários do MS-AD sejam provisionados no servidor de diretório Oracle Internet Directory (OID) e, além disso, permite delegar a autenticação ao MS-AD.

1.1 Objetivo do Trabalho

O objetivo desse trabalho é realizar um estudo das tecnologias disponíveis para sincronização de diretórios com outras fontes de dados, pois conforme constatado a tendência em muitas organizações é a utilização do MS-AD como peça central da organização.

Este trabalho se propõe também criar um protótipo de um sistema para sincronização do MS-AD com informações de empregados de um sistema de recursos humanos hipotético armazenado num banco de dados Microsoft SQL Server (MS-SQL), porém será mostrado que apesar da implementação ter sido com banco de dados, facilmente poderia ter sido outra fonte de dados.

Esse trabalho está dividido da seguinte forma: No Capítulo de Conceitos Básicos são abordados conceitos que serão utilizados ao longo trabalho, estes conceitos incluem serviço de nomes, diretórios, LDAP etc. No Capítulo de Ferramentas Disponíveis no Mercado analisa as ferramentas para integração do MS-AD com outros diretórios e repositórios. No Capítulo de Protótipo é implementado um protótipo em Java do sistema intitulado DirSync que realiza o provisionamento, desprovisionamento, atualização e desativação de usuários do MS-AD baseado numa base de dados do sistema de recursos humanos. E no capítulo de Conclusão resumimos os principais resultados do trabalho, com uma visão geral do que foi apresentado e possíveis trabalhos futuros.

2 Conceitos Básicos

Nesse capítulo serão vistos vários conceitos fundamentais para estudo das tecnologias disponíveis do mercado e para a elaboração do protótipo. Os três conceitos principais são os Serviços de Nomes, Serviços de Diretórios e o protocolo LDAP.

2.1 Serviço de Nomes

Segundo Pitágoras "Tudo são números". No que se diz respeito a computadores essa frase se aplica perfeitamente. Neles, tudo é manuseado em forma de números, sejam as letras, os endereços IP, arquivos, posições de memória, etc. Sabemos que os seres humanos memorizam nomes com bem mais facilidade do que números.

Assim, uma peça fundamental em sistemas computacionais é o serviço de nomes – que basicamente associa um nome a números ou referências. Facilitando o acesso aos objetos, que dessa forma são encontrados através apenas do nome. Sistemas típicos incluem DNS, Sistema de arquivos, etc.

Servidores de nomes armazenam suas informações de forma hierárquica. Por exemplo, para desvendar o endereço IP de www.google.com.br começa-se a partir do elemento *br* a partir daí parte-se para o elemento *com* que por sua vez possui os elementos *google* e *www*. O www é o elemento raiz e seu valor é o IP do host de destino.

2.2 Serviço de Diretórios

Atualmente, pessoas e negócios dependem cada vez mais da computação distribuída. Para facilitar o uso, reduzir os custos da administração das informações sobre os serviços, recursos, usuários, impressoras, máquinas, etc. as informações precisam ser organizados de uma maneira simples e consistente. Várias dessas informações devem ser compartilhadas por vários aplicativos, mas não devem permitir acesso ou modificação a usuários não autorizados.

Essas informações são normalmente organizadas em forma de diretórios. Serviços de Diretórios estendem de certa forma o conceito de Serviço de Nomes, pois o servidor de nomes associa um nome a apenas um valor, enquanto que em diretórios

um nome pode estar relacionado a um objeto que por sua vez possui vários atributos, onde cada atributo está associado a um ou vários valores.

Como os diretórios são distribuídos através da rede, protocolos especializados são necessários para consultar e manipular os diretórios remotamente.

Diretórios e bancos de dados tem várias características em comum como a característica de permitir a manipulação dos dados, realizarem o controle de acesso, etc.

Porém, os diretórios foram projetados para ser “read-mostly”, e são otimizados para a operação de leitura, mas isso não significa que os dados não podem ser alterados. Por exemplo, clientes de impressão podem procurar no diretório por impressoras que satisfaçam algum critério de velocidade de impressão ou local mais conveniente. Porém, as informações da impressora raramente mudam e assim não é uma tarefa comum a atualização em diretórios.

Além disso, os dados nos diretórios são armazenados de forma hierárquica enquanto que num banco de dados é uma estrutura relacional. E segundo [IBMLDAP] raramente os diretórios suportam transações ACID (Atomicity, Consistency, Isolation e Durability) que sistemas OLTP (Online Transaction Processing) possuem.

2.3 LDAP

Da necessidade de acessar diretórios, nasceu o X.500, que é um padrão para acesso a serviços de diretório rodando em cima da pilha de protocolos OSI. O X.500 definia como protocolo de comunicação entre o cliente e o servidor o DAP (Directory Access Protocol) que foi logo substituído pelo LDAP (Lightweight Directory Access Protocol), que é um padrão aberto da indústria e como o nome sugere é bastante simples se comparado com o DAP. LDAP está hoje na versão 3, está definido na [RFC2251] e provê uma forma de buscar e atualizar de informações em diretórios que rodam sobre TCP/IP.

LDAP requer a pilha de protocolos mais leve TCP/IP ao invés da pilha de protocolos OSI. LDAP simplifica algumas operações e omite algumas funcionalidades não utilizadas do DAP.

Servidores LDAP podem ser distribuídos, e a informação contida no diretório pode ser particionada ou replicada. Quando a informação é particionada, cada diretório armazena um ramo da árvore disjunto de outro ramo em outro diretório. Quando a informação é replicada, a mesma entrada do diretório LDAP é armazenada em vários diretórios. A replicação é uma funcionalidade nativa e fácil de configurar.

Várias grandes empresas participaram da elaboração do LDAP, entre elas: Universidade de Michigan, IBM, Lotus, Netscape, Microsoft. O LDAP foi reconhecido como um padrão da IETF (Internet Engineering Task Force) e está definido em várias RFC's. O LDAP é implementado por diversas empresas, entre elas:

- Apache (através do Apache Directory Server)
- Apple (através do Open Directory/OpenLDAP)
- AT&T
- Banyan
- eB2Bcom (através do View500)
- Hewlett-Packard
- IBM/Lotus
- ISODE (através do M-Vault server)
- Microsoft (através do Active Directory)
- Netscape (agora em Sun Microsystems and Red Hat products)
- Novell (através do eDirectory)
- OctetString (através do VDE server)
- Oracle (através do Oracle Internet Directory)
- Radiant Logic (através do RadiantOne Virtual Directory Server)
- Red Hat (através do Red Hat Directory Server)
- Siemens AG (através do DirX server)
- SGI and

- Sun (através do iPlanet and Sun ONE directory servers)
- Symlabs (através do Directory Extender)

O LDAP pode ser descrito como uma combinação dos seguintes modelos:

- Modelo da Informação: Descreve a estrutura da informação na árvore.
- Modelo de Nomes: Descreve como a informação é organizada e referenciada.
- Modelo Funcional: Descreve o que pode ser feito com a informação.
- Modelo de Segurança: Descreve como a informação é protegida na árvore.

2.3.1 Modelo de Informação

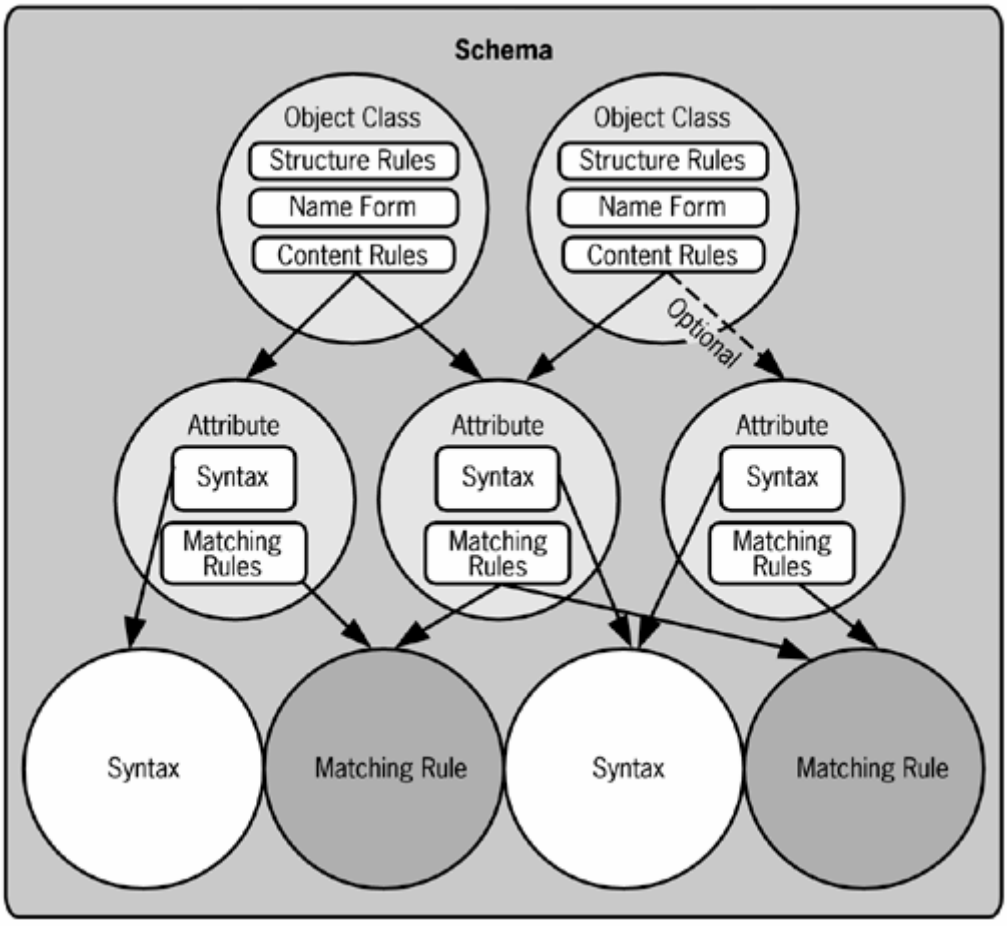

Figura 3 - Esquema do LDAP Fonte: [LDAPExplained]

O modelo de informação descrito na [RFC1777] e [RFC2251] pode ser visto na Figura 3 que define as seguintes entidades:

- Esquema: Funciona como um template para o diretório. O esquema define o conjunto de classes, atributos e outras regras que especificam quais entradas são permitidas no LDAP..
- Classes: Uma classe de objetos define o tipo de entrada permitida no diretório. Uma definição de classe consiste em regras de conteúdo, regras de estrutura, a forma de nome e alguma informação operacional adicional. Regras de conteúdo especificam quais atributos que a classe contém. Regras de estrutura define onde instâncias dessa classe podem residir na DIT. A forma de nome define quais atributos podem ser utilizados para nomear entradas daquela classe.
- Atributos: São utilizados para descrever as classes definidas no esquema. São definidas no esquema separadamente das classes para permitir que uma única definição de um atributo seja reutilizada em diversas classes. Um atributo é definido pela sintaxe, regras de comparação e algumas informações operacionais adicionais.

Classe é uma categoria de objetos que compartilham um conjunto de características. Cada objeto no diretório é uma instância de um ou mais classes do esquema, através da utilização do atributo *objectClass*. O atributo *objectClass* define a classe que um objeto pertence. Cada classe define os atributos obrigatórios e opcionais de uma dada entrada daquela classe.

Na Tabela 1 abaixo, tem-se um exemplo da definição da classe *person* que herda da classe *top*, e utiliza os atributos *cn* e *sn* que foram definidos no esquema da classe.

```
cn ( 2.5.4.3 NAME 'cn' SUP name )
sn ( 2.5.4.4 NAME 'sn' SUP name )
description ( 2.5.4.13 NAME 'description' EQUALITY caseIgnoreMatch
      SUBSTR caseIgnoreSubstringsMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{1024} )
...

person
   ( 2.5.6.6 NAME 'person' SUP top STRUCTURAL MUST ( sn $ cn )
     MAY ( userPassword $ telephoneNumber $ seeAlso $ description ) )
```

Tabela 1 - Definição de alguns atributos e da classe person

No esquema acima, foram definidos três atributos. Na definição de cada atributo, tem-se um número hierárquico seguido do tipo e algumas propriedades. Logo depois se tem a definição da classe *person* que herda da classe *top* e tem como atributos obrigatórios (*sn,cn*) e define alguns atributos opcionais (*description, seeAlso, telephoneNumber, userPassword*)

- Sintaxe: A sintaxe de um atributo define quando um determinado objeto deve ser string, número, data e assim por diante.
- Regras de comparação definem como comparar valores em operações LDAP.
- Entradas: Pode ser um container ou um objeto folha de uma determinada classe. Uma entrada é composta de vários atributos que estão na definição da classe. Cada entrada deve possuir um RDN (Relative Distinguished Name) que deve ser único entre as entradas irmãs e a concatenação dos RDNs forma o DN (Distinguished Name) deve ser único para toda árvore.

Pode-se utilizar LDAP para armazenar qualquer tipo de informação, desde que o objeto seja descrito em termos de vários atributos. Por exemplo, pode-se armazenar:

- Empregados: Nome completo, login, senha, endereço, telefone, cargo, email, cargo, empresa, etc.
- Impressoras: Nome, IP, localidade, velocidade, etc.

Como o diretório LDAP pode ser customizado para armazenar qualquer tipo de informação, basta definirmos a classe apropriada para armazenar o tipo desenado. A API de Java denominada JNDI [JNDI] permite que seja salvo qualquer tipo de objeto Java no diretório LDAP através da extensão do esquema. Diretórios LDAP armazenam a informação de uma entrada através de uma série de pares (nome, valor), uma entrada do LDAP seguindo o formato LDIF especificado em [RFC2849] seria:

```
dn: cn=Barbara Jensen, ou=Product Development, dc=airius, dc=com
objectclass: top
objectclass: person
cn: Barbara Jensen
sn: Jensen
uid: bjensen
```

```
telephonenumber: +1 408 555 1212
description: A big sailing fan.
```

Tabela 2 - Entrada LDAP

Na Tabela 2 a entrada do LDAP representa uma pessoa (classe person), note novamente que o atributo *objectClass* é multivalorado. Apesar de nesse exemplo, o atributo *description* possuir apenas um valor, em sua definição *description* é multivalorado.

2.3.2 Modelo de nomes

Conforme citado anteriormente, as entradas são identificadas unicamente através do Distinguished Name (DN), que consiste em diversos Relative Distinguished Name (RDN). Um *RDN* é semelhante a um nome de diretório ou arquivo enquando que o *DN* é semelhante a um caminho completo de um arquivo num sistema de arquivos. Pode ser utilizado qualquer atributo com valor único no RDN.

Os servidores LDAP armazenam suas informações hierarquicamente. A raiz da árvore de diretórios é semelhante à forma *dc=cin, dc=ufpe, dc=br*, que é herdado dos componentes DNS da empresa. Debaixo do elemento raiz, tem-se normalmente containers para separar os dados e por questões históricas segundo [INTLDAP] as empresas utilizam OU (Organizational Unit) para representar diferentes departamentos da empresa como: Departamento Financeiro, Departamento Comercial, Secretaria, etc.

Por exemplo, uma entrada no diretório LDAP teria DN:

cn=sscf,ou=2002-1,ou=alunos,dc=cin,dc=ufpe,dc=br. Uma lista de abreviações dos atributos segue na tabela abaixo:

Abreviação	**Significado**
uid	User id
cn	Common Name
sn	Surname
l	Location
ou	organizationalUnit

o	Organization
dc	Domain Component
st	State
c	Country

Tabela 3 - Abreviação dos Atributos LDAP

A Figura 4 ilustra um esquema hipotético de um subconjunto do MS-AD do Centro de Informática (CIn)

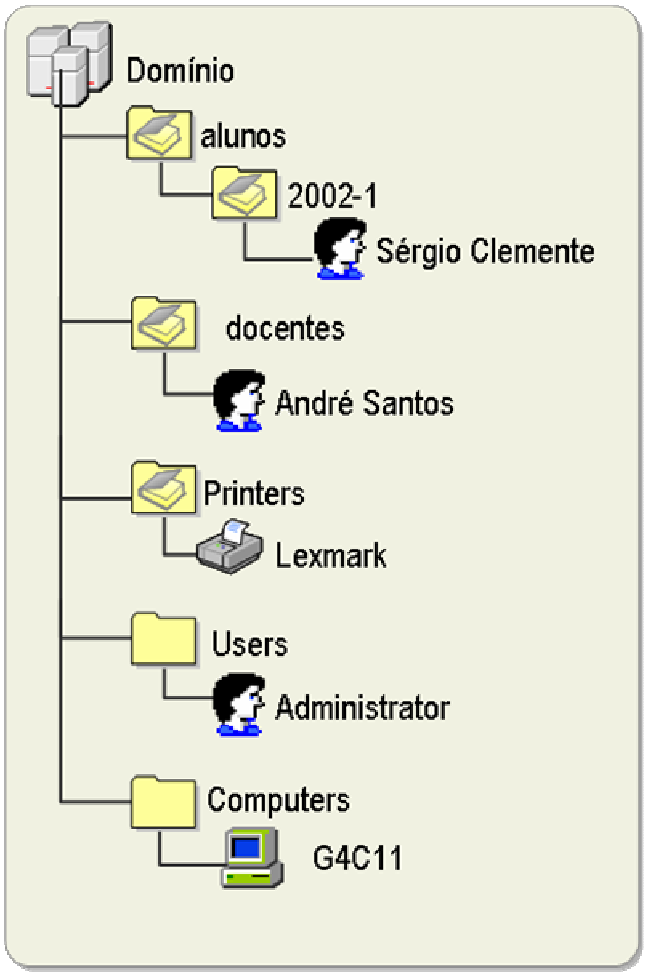

Figura 4 - Hierarquia do LDAP

Conforme constatado, o MS-AD é organizado de hierárquica e reflete estruturas políticas, geográficas ou organizacionais. Note que são armazenadas informações de impressoras, usuários, computadores, etc.

2.3.3 Modelo Funcional

O modelo funcional contempla operações em três áreas:

- Autenticação: Permite que um cliente prove sua identidade ao DSA (Directory System Agent).
- Busca: Permite que um cliente consulte o diretório. Quando um cliente realiza uma busca na DIT do diretório, o servidor retorna com uma resposta ou com um ponteiro (referral) referenciando outro servidor LDAP.
- Atualização: Permite que o cliente altere ou adicione entradas na DIT do diretório.

2.3.3.1 Autenticação

A autenticação inclui as seguintes operações:

- Open: Cria e inicializa uma conexão com o Directory System Agent (DAS).
- Bind: O comando inicia um protocolo de sessão com o das e após a sessão ser estabelecida, algum método de autenticação é negociado entre o DSA e o cliente, depois que o cliente é autenticado pelo DSA, que retorna um resposta do tipo Bind para o cliente.
- Unbind: Esse comando termina a sessão LDAP entre o cliente e o DSA.

2.3.3.2 Busca

A busca contempla as seguintes operações:

- Busca: Utilizado para procurar entradas num contexto específico da DIT do diretório, baseado nos seguintes argumentos:
    - Início da busca: Distinguished Name do objeto base de onde a busca começará.
    - Escopo da busca: Diz o quão profundo deverá ser realizada a busca. As seguintes opções estão disponíveis:
        - ✓ Base: Procura apenas no objeto atual.
        - ✓ Um nível: Procura nos objetos imediatamente abaixo do objeto atual, porém sem incluir o objeto atual.

- ✓ Sub-árvore: Procura toda sub-árvore abaixo do objeto atual, incluindo o objeto base.

- Filtro: Permite retornar apenas os objetos que satisfizerem à condição especifidada. É semelhante à cláusula WHERE do SQL, porém a notação é prefixada ao invés de infixada. Por exemplo, em vez da condição ser *cidade='SP' AND idade > 10*, o filtro é *(&(cidade=SP)(idade>10))*.

2.3.3.3 Atualização

Permite manipular as entradas, contém as seguintes operações:

- Adicionar: Adiciona uma nova entrada em uma determinada região DIT.
- Modificar: Modifica todos ou alguns atributos de uma entrada LDAP.
- Remover: Remove uma entrada.
- Renomear DN: Renomeia ou move uma entrada do LDAP. Semelhante a operação de renomear e mover um arquivo.

2.3.4 Modelo de Segurança

O modelo de segurança especifica como acessar a informação no diretório de uma forma segura. A [RFC2251] estabelece que mecanismos de SASL (Simple Authentication and Security Layer) devem ser utilizados com LDAP para prover serviços seguros de associação.

O diretório LDAP raiz possui um atributo chamado supportedSASLMechanisms que é uma lista de funcionalidades SASL suportadas.

O SASL descreve um método para prover suporte a autenticação em protocolos baseado em conexões. Para utilizar essa especificação, protocolos precisam incluir um comando para identificar e autenticar usuários num servidor e, opcionalmente, negociar proteção das interações subsequentes. Se a proteção for negociada, uma camada segura (security layer) é inserida entre o protocolo e a conexão. Semelhante a forma que o HTTPS funciona. Por exemplo, o Active Directory suporta o mecanismo SASL através dos algoritmos Kerberos 5 e o MS Negociate.

3 Ferramentas Disponíveis no Mercado

3.1 Centrify DirectControl

Centrify DirectControl permite um ambiente seguro e conectado integrando de forma transparente seu ambiente Linux, Unix, Mac, Java e plataformas web com o MS-AD. O Centrify DirectControl não requer reconfigurações de ambientes existentes e provê uma forma de gerenciamento única para um conjunto bastante diverso de sistemas e aplicações.

O funcionamento do Centrify DirectControl é bastante simples e pode ser visto na Figura 5. Em cada máquina cliente (Debian Linux, HP-UX, Novell SUSE Linux, Red Hat Linux, Solaris, Mac OS) é instalado o cliente DirectControl. Esse programa fazer com que cada cliente se comporte como um cliente Windows, gerando todo tráfego LDAP e Kerberos necessário entre o cliente e o MS-AD.

Numa máquina Windows qualquer (não necessariamente o servidor) deve ser instalado DirectControl Management Tools. Esta é a ferramenta de gerenciamento que permite gerenciar os usuários do Windows, Unix/Linux, e assim como gerar relatórios, etc. O administrador na ferramenta DirectControl Management Tools pode manipular as propriedades das contas dos usuários.

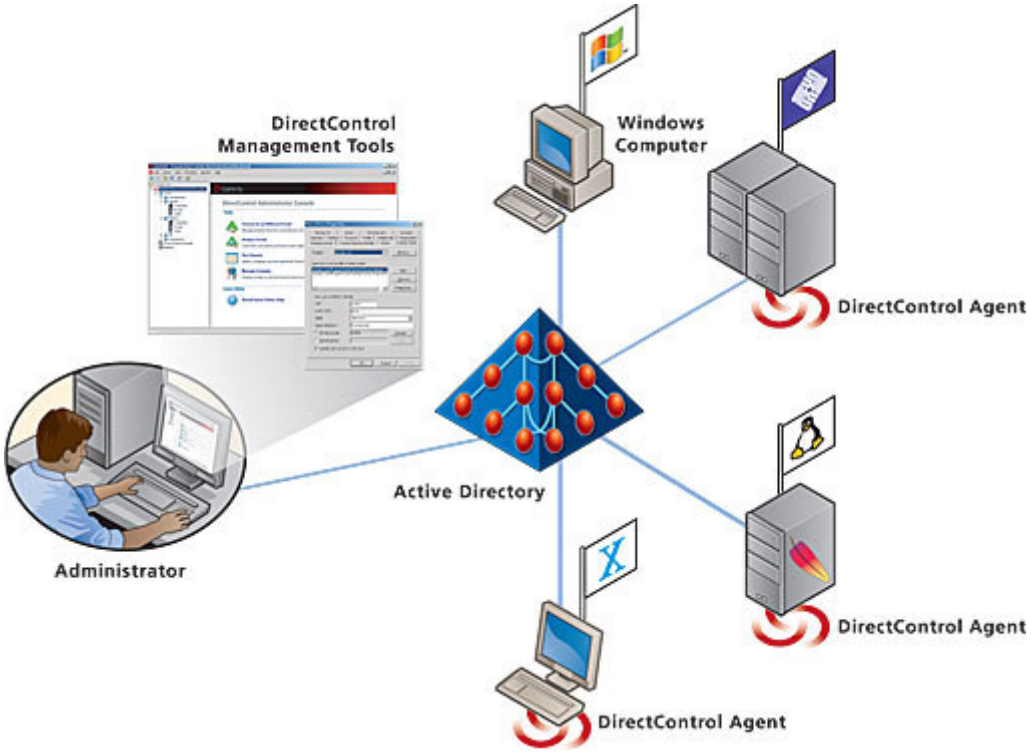

Figura 5 – Visão geral do Centrify

A implementação do DirectControl é bastante complexa. O interessante é que o esquema do MS-AD não é estendido como é a maioria das outras soluções como o Microsoft Exchange Server, Microsoft ISA Server, Microsoft Services For Unix. Estender o esquema significa definir novos atributos, e adiciona-los a classes existentes.

O DirectControl em cada cliente é composto por dois componentes: Direct Control Agent e o service library que fornece a funcionalidade a todos componentes. O Direct Control Agent roda como um daemon e gera todo tráfego LDAP e Kerberos entre o host cliente e o MS-AD.

Resumindo Centrify Precisa implementar toda uma infra-estrutura para servir as aplicações da camada superior serviços de autenticação. Ou seja, quando um aplicativo J2EE for autenticar o usuário, ele vai chamar transparentemente uma API do service library que vai se encarregar de se comunicar com o MS-AD.

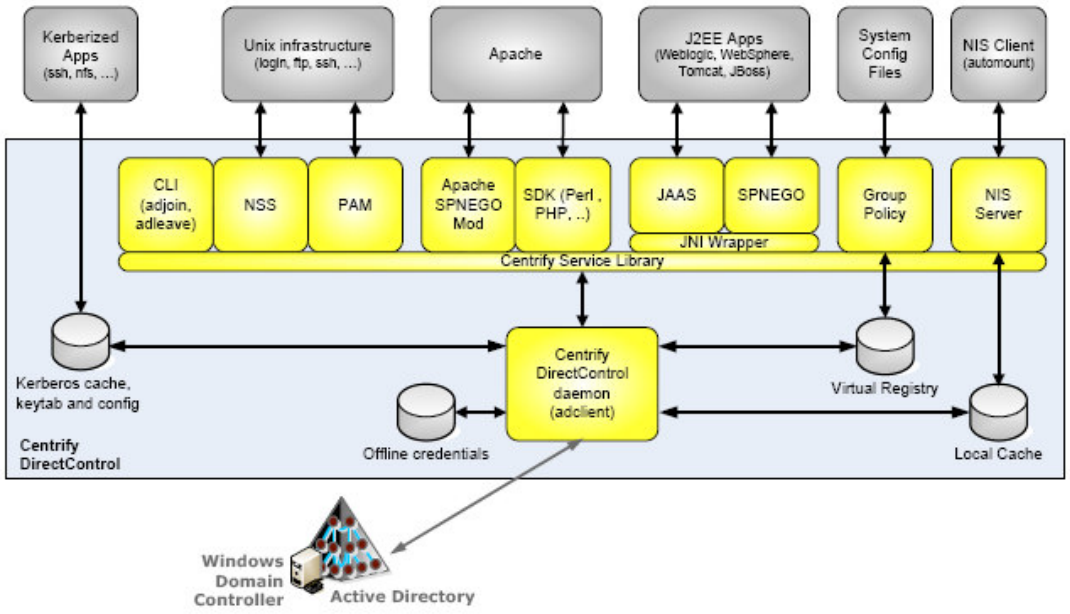

Figura 6 - Arquitetura do DirectControl

A principal desvantagem do Centrify é a necessidade de instalar um software em cada host cliente. Além disso, DirectControl não se integra com o Oracle Application Server e Oracle Internet Directory. O DirectControl Agent só funciona no Apache HTTP Server, Tomcat, JBoss AS, BEA WebLogic, IBM Websphere.

3.2 Microsoft Services for Unix (SFU)

Se a Microsoft não consegue superar absolutamente o Unix/Linux, pelo menos pode melhorar a interoperabilidade entre esses sistemas. Foi assim que surgiu o Microsoft Windows Services for Unix (SFU) [SFU], que permite computadores Windows e Unix like compartilhem dados, credenciais de segurança e scripts. SFU permite que aplicativos e scripts Unix rodem nativamente no Windows com alto desempenho.

O SFU provê o suporte ao protocolo, ferramentas de interoperabilidade, ambiente de execução e um framework de administração com objetivo final de tornar o mais simples possível a integração entre esses dois ambientes.

Possui as seguintes funcionalidades:

- Compartilhamento de arquivos: SFU suporta o sistema de arquivo do Unix Network File System (NFS) versão 2 e 3.
- Acesso remoto a linha de comando para tanto Windows quanto Unix.

- Administração de uma infraestrutura em comum: Os objetos do Network Information System[3] (NIS) são armazenados no MS-AD, assim usuários e grupos do Unix são administrados de uma forma idêntica aos objetos do MS-AD.
- Permite a sincronização uni e bidirecional dos usuários entre Windows e Unix.
- Habilidade de executar scripts na plataforma windows
- Administração dos atributos do NIS através do Microsoft Management Console (MMC) ou via linha de comando.

A instalação do SFU estende o esquema do MS-AD adicionando novos atributos, esses atributos são citados a seguir:

- NIS Domain: Domínio do NIS
- UID: É um identificador que quando configurado para a forma interativa é incrementado automaticamente.
- Login Shell
- Home directory
- Primary group/GID

A Microsoft não cobra nada pelo SFU versão 3.5, o intuito foi de facilitar a migração do Linux para Windows. O que aconteceu na prática foi facilitar um ambiente misto e efetuar a migração no sentido contrário.

Acredita-se que como resposta ao que foi citado acima, segundo a [SFU2] a Microsoft não lancará a versão 4.0 do SFU, e, além disso, o suporte do SFU estará restrito até 2011. Os componentes primários do SFU serão inclusos nas próximas versões do Sistema Operacional.

---

[3] Network Information Service, um serviço que fornece informação que deve ser conhecida pela rede, para todas as máquinas da rede. Após algum tempo foi lançado o NIS+, como uma substituição para NIS com melhor segurança e melhor usabilidade para grandes instalações.

3.3 MIIS

O Microsoft Identity Integration Server (MIIS) é um serviço centralizado que armazena e integra informação de identidade para organizações com múltiplos diretórios. O objetivo do MIIS é prover uma visão unificada de todos os usuários, aplicações e recursos computacionais.

O MIIS permite sincronizar contas de usuários com diversas fontes de dados como outros diretórios ou estruturas heterogêneas como arquivos texto e banco de dados, por exemplo. A sincronização inclui provisionamento (criação), desprovisionamento (exclusão) e atualização de contas.

O MIIS permite a sincronização de senhas e um Aplicativo Web permite que através de um único clique do usuário, as senhas sejam atualizadas em múltiplas fontes de dados.

Apesar de atender a todas as necessidades, o grande problema do MIIS é o preço da licença. Na época da elaboração deste trabalho o custo da licença por processador do MIIS era $24.999, mais a licença do Windows Server 2003 Enterprise Edition e do SQL Server que são necessárias para o funcionamento do MIIS que juntos custam no mínimo $ 5.848 (Que depende da forma de licenciamento escolhida) resultando em $30.847.

Como será explicado posteriormente, o MIIS é uma plataforma de desenvolvimento, assim terá que ser incluso nessa lista ainda o custo de desenvolvimento associado. Por esses e outros motivos que o MIIS não vem sendo adotado em larga escala.

3.3.1 Como funciona?

A Figura 7 contém a arquitetura do MIIS. Os componentes principais são: Connected Data Source, Management Agents, Connector Space e o Metaverse.

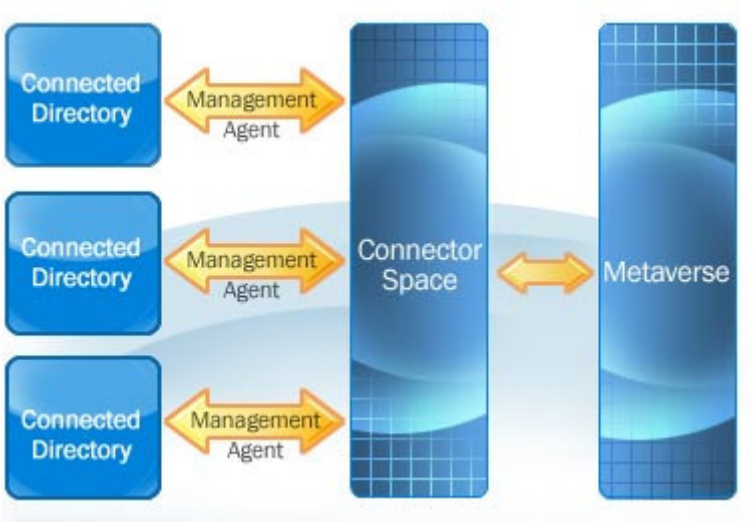

Figura 7 - Como o MIIS Funciona. Fonte [MIIS]

3.3.2 Connected Data Source

Um Connected Data Source pode ser um diretório, um banco de dados, ou qualquer outro repositório que contenha dados para ser integrado com o MIIS como arquivos texto, diretórios de e-mail, arquivos XML, etc.

3.3.3 Management Agents

O Management Agents conecta uma fonte de dados com o MIIS. É responsável por integrar os dados entre o MIIS e o Connected Data Source. Quando os dados no MIIS são modificados, o Management Agent pode refletir essas mudanças nas Connected Data Source's respectivas para manter a integridade. Em regra geral existe um Management Agent para cada fonte de dados conectada. O MIIS 2003 inclui agentes de gerenciamento para as seguintes fontes de dados:

- Active Directory
- Active Directory Application Mode (ADAM)
- Attribute-value pair text files
- Comma separated value files
- Delimited text files
- Directory Services Markup Language (DSML) 2.0
- Exchange 5.5, Exchange 5.5 Bridgehead

- Exchange 2000 and Exchange 2003 Global Address List (GAL) synchronization
- Fixed-width text files
- IBM DB2, IBM Tivoli Directory Server
- LDAP Directory Interchange Format (LDIF)
- Lotus Notes/Domino 4.6/5.0/6.0
- Novell eDirectory
- Sun/iPlanet/Netscape directory 4.x/5.x (with “changelog” support)
- Microsoft SQL Server 2000, SQL Server 7.0
- Microsoft Windows NT 4 Domains
- Oracle 8i/9i
- Informix, dBase, ODBC and OLE DB support via SQL Server Data Transformation Services

Management Agents contêm regras de como os objetos são mapeados, quando os objetos da fonte de dados conectada devem ser criados ou excluídos.

Um grande benefício da implementação do MIIS é que nenhum software adicional precisa ser instalado no Connected Data Source, o que difere do Centrify DirectControl. Os dados de identidade são armazenados em dois namespaces lógicos: Connector Space e o metaverse.

3.3.3.1 Connector Space

O Connector Space (CS) é uma área de armazenamento que é utilizado pelos Management Agents para mover os dados para ou da Connected Data Source. O CS é essencialmente um espelho do dado das fontes de dados conectadas relacionada, existindo uma correspondência biunívoca entre cada elemento da fonte de dados conectada e cada elemento do CS.

O CS mantém um registro do estado do objeto e dos seus atributos. Quando estiver acontecendo à sincronização com a fonte de dados o CS vai conter tanto o estado

atual, quanto o estado antigo do atributo do objeto assim como informações se o registro foi é criado, apagado ou atualizado.

Dependendo da lógica de negócio, uma mudança no estado atual do objeto pode implicar numa desativação de um objeto do MS-AD.

O CS provê transparência total para todas as operações que acontecem com um objeto, independente de ser inclusões, exclusões e atualizações.

3.3.3.2 Metaverse

O metaverse é o responsável por integrar as diversas fontes de dados, toda informação de uma pessoa, que é armazenada em múltiplos CS são resumidos a apenas um metaverse.

Quando um agente de gerenciamento é executado, modificações que forem feitas nas Connected Data Sources são refletidas no Connector Space e as regras são aplicadas e os dados resultantes são preenchidos no metaverse. O metaverse manda as modificações ao espaço conectado de outras fontes de dados para propagar as modificações.

3.4 Oracle Directory Integration and Provisioning Platform (DIP)

A plataforma de integração do oracle permite a sincronização entre o Oracle Internet Directory (OID) e outros servidores LDAP como, o MS-AD, SunOne Directory através do Directory Integration and Provisioning Server (DIP Server).

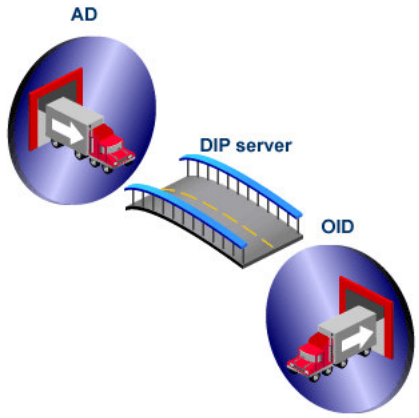

Figura 8 – DIP Server

Para a integração com o MS-AD o Oracle permite:

- Realizar a sincronização uni ou bidirecionalmente.
- Especificar como os atributos serão mapeados entre os diretórios LDAP.
- Sincronizar com vários servidores MS-AD. Pode-se sincronizar diretamente com um único servidor ou com toda a floresta de domínios do MS-AD através do Microsoft Global Catalog.
- Utilizar a autenticação do Windows, ou seja, quando o ambiente Oracle necessitar autenticar o usuário, a autenticação realizada pelo Kerberos do MS-AD.

Um detalhamento da figura anterior é dado pela Figura 9:

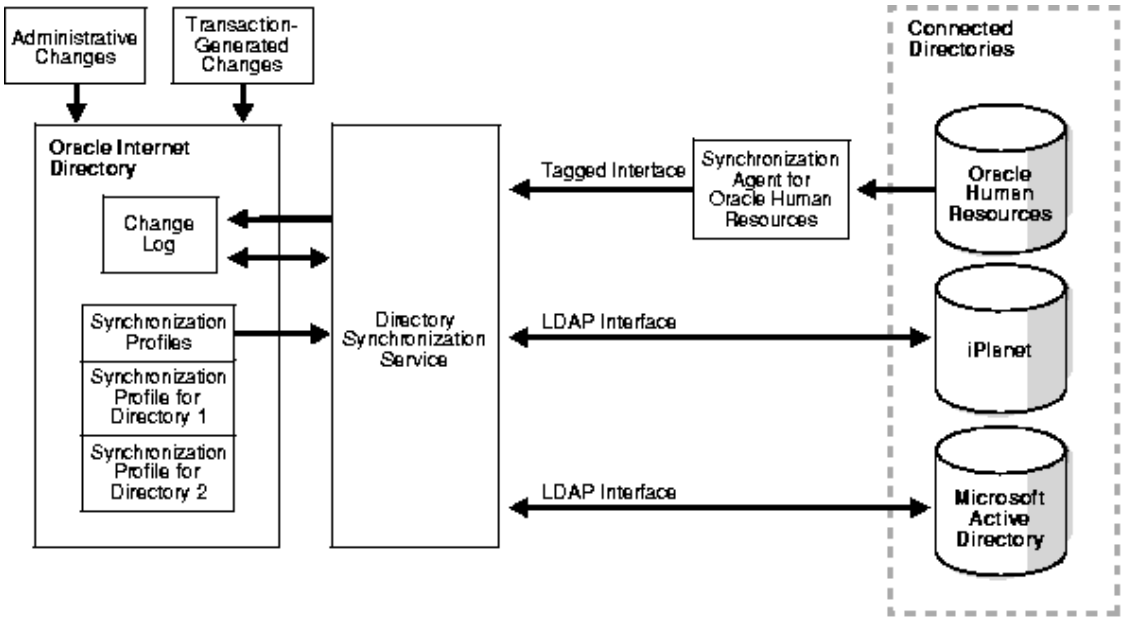

Figura 9 - Componentes da plataforma de integração da oracle

Note que o Oracle permite a integração com o recursos humanos, porém não se recomenda utilizá-lo, pois conforme visto em [Centrify2] a tendência tem sido centralizar as modificações no MS-AD e não no OID.

Um guia completo de como sincronizar o MS-AD com o OID pode ser visto em [OracleIntegration2]. Em linhas gerais são realizados as seguintes atividades:

- Criar arquivos de mapeamento de atributos e containers e carregar no ActiveChgImp profile. O profile ActiveChgImp é o agente responsável por sincronizar os usuários do MS-AD no OID.
- Dar permissão para criação dos grupos do MS-AD no OID: Ao contrário de outros diretórios LDAP, o MS-AD armazena grupos no container *Users*. Assim, faz-se necessário a criação de uma access control policy para permitir a criação de grupos no container *Users*.
- Entrar no Oracle Internet Directory e configurar o ActiveChgImp: Consiste em configurar o agente que irá realizar a sincronização do MS-AD para o OID. Os seguintes dados deverão ser preenchidos no ActiveChgImp:
    - Período para repetição do agente: De quanto em quanto tempo o agente deverá sincronizar os usuários.
    - Nome completo do servidor LDAP (FQDN), login e senha do usuário que será utilizado para logar no diretório LDAP.
    - Filtro de entradas do LDAP que não serão sincronizadas: Pois caso não especificado, as conta das máquinas serão criadas automaticamente no OID.
    - Número da última atualização do MS-AD: O OID utiliza o atributo *highestCommittedUSN* do MS-AD para detectar quando houve alguma atualização no MS-AD não repassada para o OID ainda. Ou seja, se o *highestCommittedUSN* do MS-AD estiver igual a 100 e o valor que estiver armazenado no OID for 99 então o profile ActiveChgImp deverá efetuar uma rodada de sincronização dos usuários e grupos.
- Iniciar o DIP server: Iniciar o servidor de integração dos diretórios.
- Habilitar o ActiveChgImp
- Instalar o plugin de delegação de autenticação: Rodar o script *oidspadi.sh*. Se estiver em ambiente windows, será necessário o cygwin para rodar o arquivo.

- Habilitar o pluggin de delegação de autenticação do Oracle: Basta executar duas modificações no servidor LDAP que estão no formato LDIF para habilitar o plugin de delegação do oracle.

Após essas configurações, o ambiente Oracle irá importar automaticamente os usuários do MS-AD e autenticar os usuários no MS-AD.

Apesar da configuração parecer ser simples vários problemas podem surgir durante a configuração do ambiente. Por exemplo, os seguinte problemas aconteceram com o autor desse trabalho no ambiente Oracle 10g release 1:

- Erro na porta: O guia passo a passo ensina a iniciar o dipserver para se conectar com o OID na porta 3060, porém o OID do Oracle por default roda na porta 389, que é a porta padrão do LDAP.
- O ambiente Oracle 9.0.4.1.0 vem com um bug que faz com que o ActiveChgImp entre em loop. O bug foi resolvido aplicando um patch. Esse patch só é liberado através do contato direto com o suporte da oracle, ou seja, não é disponibilizado publicamente para download na internet.
- O arquivo odisrv.bat que inicia o odiserver vem com um bug que precisa ser reparado na mão.

Por esses problemas acima citados, recomendamos a solicitação de um Service Request (Requisição de suporte) para configuração da plataforma do Oracle para integração com o MS-AD.

4 Protótipo

Conforme constatado nos capítulos anteriores o LDAP apesar de ser um padrão, resolve o problema de centralizar a informação, porém não resolve a sincronização entre diretórios ou repositório de dados.

Neste capítulo apresentaremos o prototipo do sistema DirSync, que realizará a sincronização unidirecional da base de empregados de um sistema de RH para o MS-AD. Conforme será mostrado durante este capítulo, a fonte de dados do sistema de RH poderá ser substituída por outra fonte de dados de maneira transparente. O fluxo de informação é dado pela Figura 10:

1. Quando o empregado entrar na empresa, seu cadastro terá que ser criado por alguma pessoa do setor de RH no Sistema de Recursos Humanos (SRH).
2. O SRH persiste as alterações na base de dados.
3. O DirSync lê os dados da tabela de empregado da base de dados do SRH.
4. O DirSync lê os usuários do MS-AD.
5. O DirSync compara os dois repositórios, e realiza as alterações devidas no MS-AD.

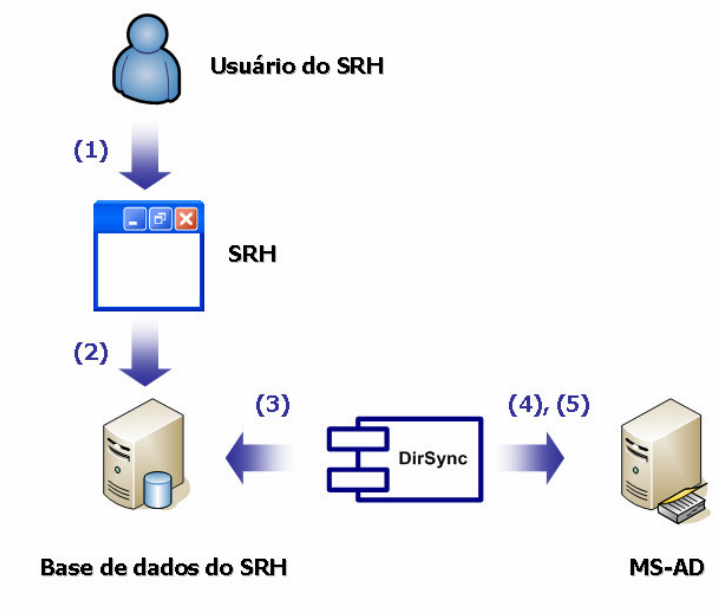

Figura 10 - Elementos macro da solução

4.1 Elementos da solução

Nesta seção será explicada mais detalhadamente a responsabilidade de cada elemento da solução.

4.1.1 Sistema de Recursos Humanos (SRH)

Aplicativo cuja função é armazenar informações de cada empregado, cuidando de monitorar planos de saude, planos de seguro, automatizar o processo de pagamento, etc. O que interessa para o protótipo é a tabela de empregados que servirá de entrada para o DirSync.

4.1.2 Usuário do SRH

O usuário da aplicação de RH é normalmente alguém do setor de RH responsável por entrar com os dados do novo empregado no sistema, assim como atualizar informações do empregado como endereço, telefone e status que informa a situação funcional do empregado. A situação funcional do empregado diz se o mesmo encontra-se em situação normal, aposentado, demitido, férias, etc. o que vai permitir inferir se um determinado empregado do SRH deverá existir ou não no MS-AD.

4.1.3 Base de dados do SRH

Contém os dados que o SRH utiliza para seu funcionamento. Numa base de dados relacional, os registros são armazenados em forma de tabelas. A tabela de empregado utilizada no protótipo está descrita na Tabela 4 abaixo:

Nome	**Descrição**	**Tipo de dado**
id	Chave primária	Número Inteiro, auto incrementado
nomeCompleto	Nome completo do empregado	Texto
telefone	Telefone	Texto
cidade	Cidade	Texto
uf	Estado	Texto
hierarquia	Hierarquia na empresa. Por exemplo: alunos/2005-1	Texto

situacaoFuncional	Código que identifica a situação funcional do empregado (normal, demitido, férias, etc)	Texto
login	Login do empregado	Texto

Tabela 4 - Tabela de empregados da base de RH

4.1.4 Microsoft Active Directory (MS-AD)

Contêm informações sobre usuários, computadores, impressoras, etc. Para o protótipo, interessa apenas as informações das contas dos usuários, que será manipulada pelo aplicativo DirSync. Os atributos dos usuários que serão contemplados estão descritos na Tabela 5 abaixo:

Nome	**Descrição**
distinguishedName	Nome identificador no domínio
samAccountName, userPrincipalName	Refere-se ao login da conta do usuário
givenName	Primeiro nome
sn	Restante do nome
telephoneNumber	Número do telefone
l	Cidade
st	Estado

Tabela 5 - Tabela referente aos atributos da conta do usuário do MS-AD

Os usuários serão criados dentro de uma unidade organizacional especifica, no caso do protótipo, todos os usuários serão manipulados dentro de *OU=RH,DC=LABORATORIO,DC=COM,DC=BR*. O motivo principal é isolar os usuários do sistema de recursos humanos com os usuários de sistema que ficam dentro do container *Users*.

Devido a este trabalho ser apenas um protótipo, um subconjunto de atributos bastante restrito do MS-AD e da tabela de empregados do RH foram escolhidos. No

sistema que foi a base para este trabalho, a visão de empregados continha 17 colunas.

4.1.5 DirSync

Aplicativo que é responsável por comparar os repositórios do SRH e do MS-AD e atualizar o MS-AD com as informações mais recentes da base de dados do SRH. Este aplicativo foi modelado para rodar em batch, onde sua execução deve ser agendada com um período fixo como um dia, por exemplo.

Já apresentamos a tabela de empregados e os atributos do MS-AD que serão tratados. Porém, para o funcionamento do DirSync precisa existir um mapeamento entre essas entidades. A Tabela 6 relaciona essas entidades:

Atributo do MS-AD	**Regra de formação**
distinguishedName	CN=*empregado.nomeCompleto* concatenado com a *OU* respectiva a empregado.hierarquia concatenado com "OU=RH,DC=laboratorio,DC=com,DC=br"
givenName	Primeiro nome de empregado.nomeCompleto
l	empregado.cidade
SAMAccountName	empregado.login
sn	Parte do nome restante de empregado.nomeCompleto
st	empregado.uf
telephoneNumber	empregado.telefone
userAccountControl	NormalAccount or AccountDisable or PasswordExpired
userPrincipalName	empregado.login concatenado com @laboratorio.com.br

Tabela 6 - Mapeamento entre o empregado e o usuário do MS-AD

O atributo *distinguishedName* conforme explicado no Conceitos Básicos, identifica unicamente qualquer entrada do LDAP. Conforme veremos a seguir, se o valor do

campo *empregado.hierarquia* for “alunos/2002-1”, a conta do usuário será criado na OU *OU=2002-1, OU=alunos,OU=RH,DC=laboratorio,DC=com,DC=br*. Note que todos os usuários serão criados a partir da OU intitulada *RH*. Pois conforme visto anteriormente, não é uma boa prática criar os usuários no container *Users* (*CN=Users,DC=laboratorio,DC=com,DC=br)* pois diversos usuários de sistemas estão contidos nela, confusão que pode gerar confusão sobre se as conta de usuários do SRH e de sistema (sqlserver, smsadmin, krbtgt).

O atributo userAccountControl contém algumas flags responsáveis pelo estado da conta do usuário. Essas flags são valores binários e deve-se utilizar a operação *OR* binária para unir vários valores. Por questões de segurança a conta do usuário será criada desabilitada no MS-AD.

4.2 Fluxo de funcionamento do DirSync

Como mencionado anteriormente o DirSync deve ser configurado ser executado periodicamente. O DirSync é composto por dois módulos principais:

- Gerador de operações: Este módulo compara os usuários do MS-AD com a tabela de empregados do SRH e gera as operações necessárias para atualização do MS-AD. Cada operação pode ser definida como uma seqüência de comandos que tem como objetivo final provisionar, atualizar, desprovisionar ou desativar a conta do usuário. Ou seja, a operação de provisionamento cria uma conta de um novo empregado no MS-AD, a operação de desprovisionamento remove a conta do MS-AD de um empregado (por exemplo, no caso de demitido) e assim sucessivamente.
- Executor de operações: Este módulo recebe como entrada as operações geradas pelo gerador de operações e executa as mesmas no MS-AD.

4.3 Gerador de operações

O gerador de operações realiza comparações com os dados do MS-AD e da tabela de empregados do repositório do SRH. Para implementação desse protótipo, será explicada a semântica das quatro operações descritas anteriormente:

- Provisionamento: Será gerada uma operação de provisionamento para os registros da tabela de empregados cujo valor no campo situacaoFuncional estiver igual a ”NORMAL” e não possuir conta no MS-AD.
- Desprovisionamento: Será gerada uma operação de desprovisionamento para as contas do MS-AD que possuírem um registro correspondente na tabela de empregados e o campo situacaoFuncional estiver com o valor “DEMITIDO”.
- Atualização: Será gerada uma operação de atualização para as contas do MS-AD que possuírem um registro correspondente na tabela de empregados e o campo situacaoFuncional estiver com o valor “NORMAL”.
- Desativação: Será gerada uma operação de desativação para as contas do MS-AD que possuírem um registro correspondente na tabela de empregados e o campo situacaoFuncional estiver com o valor “FERIAS”.

Apesar de terem sido abordadas apenas três situações funcionais (NORMAL, DEMITIDO E FÉRIAS), poder-se-ia estender facilmente para mais valores. Numa implementação real, são tratados 27 valores distintos para o campo situacaoFuncional.

4.3.1 Arquitetura do Gerador de operações

Para geração das operações o agente Gerador de Operações precisa comparar a base de empregados do SRH com as contas de usuários do MS-AD. Essa operação será executada no backend MS-Sql através de quatro stored procedures (uma para cada tipo de operação). A Figura 11 resume genericamente a visão macro de cada procedure.

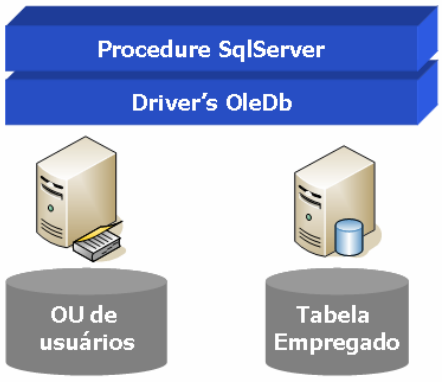

Figura 11 – Visão macro de cada stored procedure

Devido ao MS-Sql ter uma ótima integração com fontes de dados diversas, a arquitetura da Figura 11 foi utilizada. Para o leitor entender, um código Transact-Sql (dialeto Sql do MS-Sql) pode consultar qualquer fonte de dados desde que exista um driver OleDb para recuperação e manipulação dos dados desejados. Para elaboração deste protótipo será utilizado o driver OleDb para acessar fonte de dados do MS-AD. Assim, a unidade organizacional (OU) de usuários será acessada via uma consulta SQL comum. A tabela de empregados do SRH também pode ser acessada via um driver OleDb (para acessar por exemplo uma base oracle, mysql, etc), porém como no caso desse protótipo a tabela está situada no próprio MS-Sql a camada do driver OleDb não existe, conforme ilustrada na Figura 12.

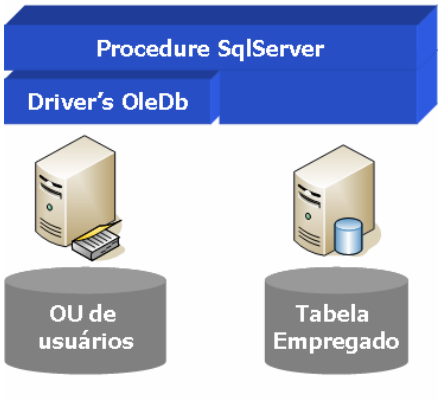

Figura 12 – Visão 2 macro de cada procedure

Note que apesar do protótipo ter integrado o MS-AD com uma tabela do banco de dados, hipoteticamente pode-se integrar com qualquer fonte de dados, desde que exista um driver OleDb para a fonte de dados desejada. Por exemplo, pode-se integrar o protótipo com arquivos do tipo comma-separated values (CSV) através do driver OleDb Microsoft Text Driver.

A criação do Linked Server pode ser feita conforme visto na Tabela 7. Deve-se informar um nome para o linked server, o endereço que está localizado e tipo de driver OleDb que será utilizado.

Logo após precisa-se informar a credencial que será utilizada para acessar o MS-AD, esse passo é feito com o comando *sp_addlinkedsrvlogin*.

Na hora da criação do linked server o usuário deverá estar logado no domínio senão o MS-Sql irá acusar um erro na execução de qualquer query do linked server.

```
exec sp_addlinkedserver @server='MSAD', @srvproduct='Active Directory
Services 2.5', @provider='ADsDSOObject', @datasrc='192.168.5.253'

exec sp_addlinkedsrvlogin @rmtsrvname = 'MSAD',
    @useself = false,
    @locallogin = null,
    @rmtuser =
'CN=Administrator,CN=Users,DC=LABORATORIO,DC=COM,DC=BR',
    @rmtpassword = 'labteste'
```

Tabela 7 - Código para criação do linked server

Após a criação do linked server, será criado a visão *v_usuarioad* que acessa as contas de usuários do MS-AD. Essa visão é dada pela Tabela 8. A visão é criada a partir de uma consulta *SELECT*, porém na cláusula *FROM* deve-se colocar o contexto de onde a consulta LDAP deve iniciar. Nesse caso a busca deve-se iniciar a partir da OU *RH* que será a OU onde todas as contas de usuários serão manipuladas.

```
create view v_UsuarioAD
as
SELECT *
FROM OPENQUERY( MSAD,
     'SELECT distinguishedName, sAMAccountName, userPrincipalName,
givenName, sn, telephoneNumber, l, st, userAccountControl
      FROM ''LDAP://192.168.5.253/OU=RH,DC=laboratorio,DC=com,DC=br''
      WHERE objectClass = ''user''')
```

Tabela 8 - View para recuperar os usuários do MSAD

Porém esse método tem uma limitação quando o atributo a ser selecionado no MS-AD é multivalorado. Pois conforme visto no Conceitos Básicos, o atributo *description* é multivalorado, e, portanto não pode ser selecionado na consulta SQL. A alternativa é acessar o MS-AD diretamente via o aplicativo (No caso de Java, pode ser utilizado a API JNDI).

A Tabela 9 ilustra o código Transact-Sql responsável pela criação da stored procedure que retorna os empregados do SRH que devem ser provisionados no MSAD.

```
create procedure sp_empregadosProvisionamento as
select emp.*
    from tb_empregado emp
    where emp.situacaoFuncional in ('NORMAL')
    and (select count(*) from v_usuarioad where sAMAccountName =
emp.login) = 0
```

Tabela 9 - Procedure Transact-Sql para recuperar os empregados que devem ser criados

A decisão de projeto de criar uma procedure foi por questões de desempenho. A rigor por ser um aspecto de lógica do aplicativo deveria ficar no código Java (ou outra linguagem utilizada na implementação). Porém quando a quantidade de dados é grande quanto mais próximo dos dados originais for feito o processamento menor tempo de resposta do algoritmo. Além disso, expressando a lógica na stored procedure pode-se utilizar de todo poder expressivo que a linguagem Sql oferece.

4.4 Executor de operações

O executor de operações recebe como entrada uma lista de operações a serem executadas e as executa no MS-AD. Caso ocorra algum erro durante a execução de uma operação, a mesma deverá salva devendo ser resumida na próxima execução do DirSync.

4.4.1 Semântica das Operações

Como foi dito anteriormente, existem quatro tipos de operações, cada tipo possui uma ou mais tarefas. Uma tarefa pode ser vista como um passo atômico.

4.4.2 Modelo de execução

Para simplificar o executor de operações, não será adotado o modelo transacional. Ao invés disso um modelo seqüencial será adotado. Assim, imagine que uma operação possui três tarefas. Caso haja um erro na segunda tarefa, o executor de operações irá salvar a operação e na próxima vez que o DirSync for executado, a operação irá recomeçar a segunda tarefa.

4.4.2.1 Provisionamento

A tarefa de provisionamento é composta por duas tarefas:

1. Criar OU's: Criar unidades organizacionais (organizational units) que reflitam a hierarquia aonde o usuário deverá ser criado. Por exemplo, se o valor do campo *hierarquia* for *alunos/2002-1*, então o usuário será criado no contexto LDAP *OU=2002-1,OU=alunos,OU=RH,DC=laboratório,DC=com,DC=br*.

2. Criar conta de usuário desabilitada: Cria a conta do usuário no diretório LDAP. A conta deverá ser criada desabilitada por questões de segurança e uma pessoa do helpdesk deverá habilitar manualmente a conta do usuário.

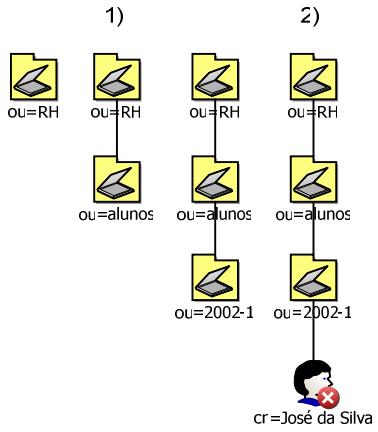

Figura 13 – Seqüência de atividades no provisionamento

4.4.2.2 Atualização

A tarefa de atualização é composta de apenas uma tarefa:

- Atualizar conta de usuário: Atualiza o valor dos atributos da entrada do LDAP.

4.4.2.3 Desprovisionamento

A tarefa de desprovisionamento é composta de duas tarefas:

- Salvar conta de usuário antiga: Salva os atributos do usuário num arquivo texto para efeitos de backup.
- Remover conta de usuário: Remove a conta do usuário do diretório LDAP.

4.4.2.4 Desativação

A tarefa de desativação é composta de apenas uma tarefa:

- Desativar conta de usuário

4.5 Detalhamento da Implementação

A implementação do protótipo foi feito em Java. A arquitetura macro da solução é dada pela Figura 14. O aplicativo DirSync acessa a base MS-Sql através do Hibernate [Hibernate] utilizando o driver JDBC para acessar o MS-Sql. Hibernate é uma API de Java que realiza a persistência objeto/relacional e provê um serviço de

consulta. O acesso ao Servidor LDAP foi possível através da API Java Naming and Directory Interface (JNDI). O JNDI é uma API de Java para acessar Servidor de Nomes e Servidor de Diretórios.

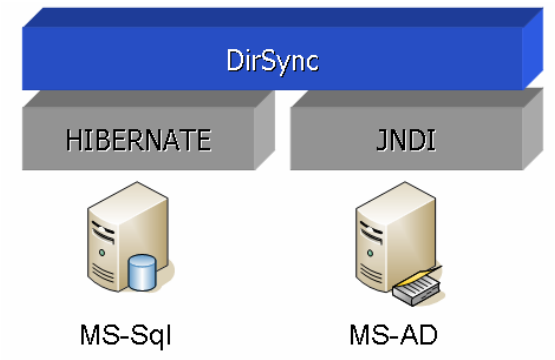

Figura 14 – Macro arquitetura do DirSync

A nomenclatura das classes utilizadas no DirSync está bastante condizente com o que foi discutido durante todo esse capítulo. O diagrama de seqüência do fluxo de provisionamento é dado pela Figura 15.

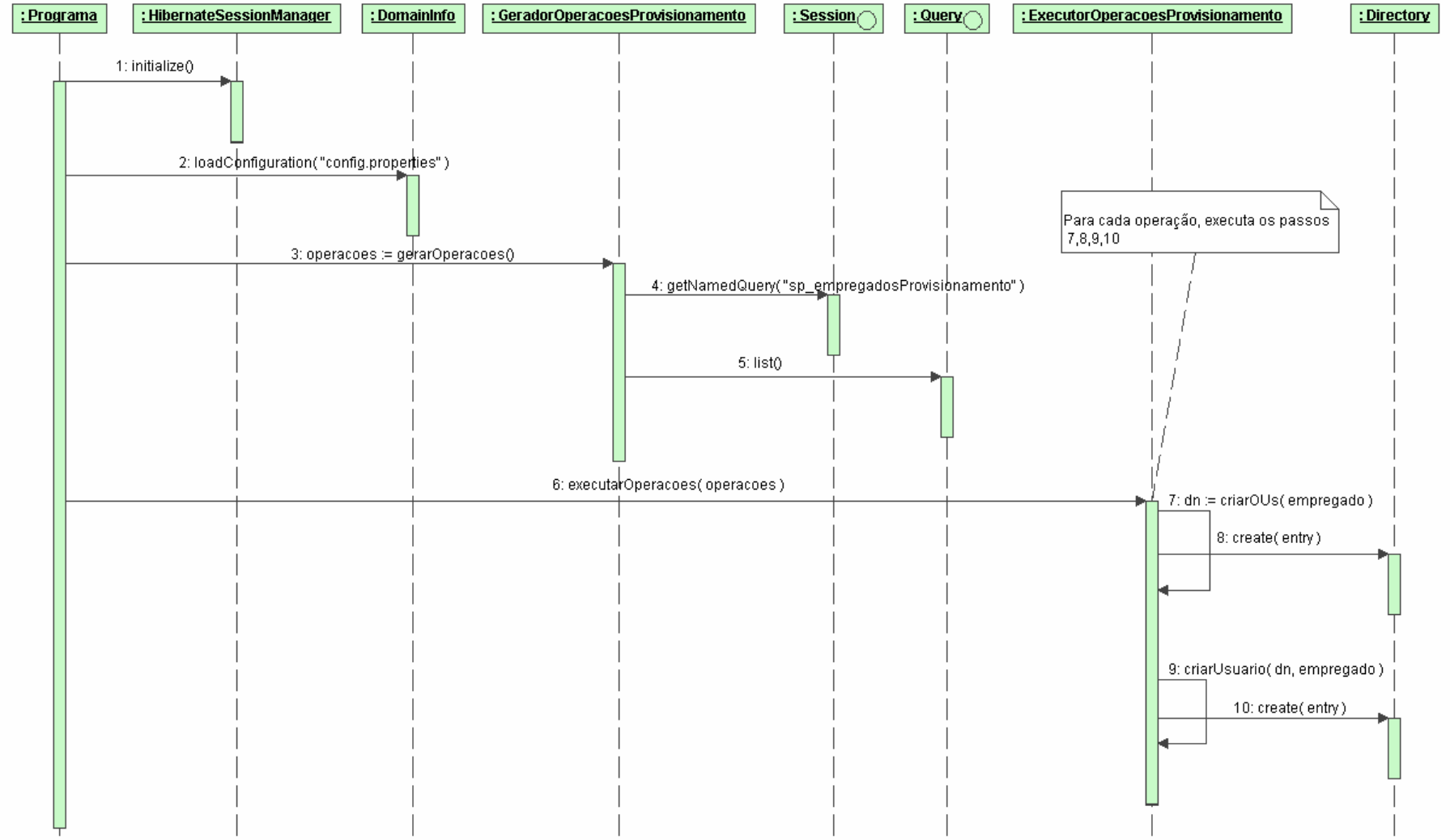
: Programa
: HibernateSessionManager
: DomainInfo
: GeradorOperacoesProvisionamento
: Session
: Query
: ExecutorOperacoesProvisionamento
: Directory
1: initialize()
2: loadConfiguration("config.properties")
3: operacoes := gerarOperacoes()
4: getNamedQuery("sp_empregadosProvisionamento")
5: list()
6: executarOperacoes(operacoes)
Para cada operação, executa os passos 7,8,9,10
7: dn := criarOUs(empregado)
8: create(entry)
9: criarUsuario(dn, empregado)
10: create(entry)

Figura 15 - Diagrama de seqüência da operação de provisionamento

Os elementos da figura são explicados a seguir:

- Programa: Contém o método main() do aplicativo Java, é disparado quando o DirSync é chamado pela linha de comando.
- HibernateSessionManager: Classe que é responsável por carregar as configurações do Hibernate e retornar novas instâncias do objeto Session. As configurações são os arquivos de mapeamento e o arquivo hibernate.hmb.xml que contém informações de como se conectar ao banco de dados como qual driver JDBC, ip, login e senha.
- DomainInfo: Classe que contém informações sobre o servidor LDAP, como ip onde está localizado, login e senha, distinguishedName do diretório (por exemplo, DC=laboratorio,DC=com,DC=br)
- GeradorOperacoesProvisionamento: Classe que contém um método para gerar as operações de provisionamento. O método gerarOperacoesProvisionamento() executa a stored procedure "sp_empregadosProvisionamento" do MS-Sql que retorna a lista de empregados que devem ser criados no MS-AD.
- Session: Classe hibernate que representa abstratamente uma conexão com o banco de dados. A classe Session contém vários métodos para consultar o banco de dados, assim como realiza caching de objetos.
- Query: classe hibernate que representa abstratamente uma consulta com o banco de dados. A consulta pode ser simples como um SELECT, porém pode ser uma stored procedure que retorne um conjunto de resultado.
- ExecutorOperacoesProvisionamento: Classe que recebe a lista de operações de provisionamento e cria as OU's e usuários no MS-AD.
- Directory: Classe que encapsula a API JNDI contém métodos para manipular o diretório LDAP, como para criar, remover, verificar se uma entrada existe, etc.

O usuário ativa o DirSync pela linha de comando, o método main() da classe Principal é invocado. O método initialize() da classe HibernateSessionManager é chamado para carregar as configurações do Hibernate. Logo após, o método loadConfiguration() é invocado para carregar as configurações do servidor LDAP.

As operações são geradas com a execução do método gerarOperacoes() da classe GeradorOperacoesProvisionamento. A lista de operações retornadas pelo gerador de operações é passada para a classe ExecutorOperacoesProvisionamento. Para cada operação da lista de operações, as unidades organizacionais (OU) necessárias são criadas, assim como o usuário é criado na unidade organizacional que reflita o campo hierarquia do empregado.

O código do método main() que foi explicado pelo diagrama de seqüência da Figura 15 é dado pela Tabela 10.

```
public static void main(String[] args) {
  try {
    // Carrega o hibernate
    HibernateSessionManager.getInstance().initialize();

    // Carrega as informações do domínio
    DomainInfo.getInstance().loadConfiguration("config.properties");

    // Gera as operações
    GeradorOperacoesProvisionamento gop = new
GeradorOperacoesProvisionamento();
    List operacoes = gop.gerarOperacoes();

    // Executa as operações
    ExecutorOperacoesProvisionamento eop = new
ExecutorOperacoesProvisionamento();
    eop.executarOperacoes(operacoes);
  } catch (Exception e) {
    e.printStackTrace();
  }
}
```

Tabela 10 - Código main()

A Tabela 11 contém o código para gerar as operações.

```
public List<OperacaoProvisionamento> gerarOperacoes() {
  List<OperacaoProvisionamento> retorno = new
Vector<OperacaoProvisionamento>();
  List listaEmpregados;

  // Recupera os usuários que devem ser provisionados
  HibernateSessionManager hib = HibernateSessionManager.getInstance();

  Session session = hib.openSession();
```

```
  Transaction tran = session.beginTransaction();

  Query q = session.getNamedQuery("sp_empregadosProvisionamento");

  listaEmpregados = q.list();

  tran.commit();

  session.close();

  // Monta as operações
  for (Object object : listaEmpregados) {
    Empregado empregado = (Empregado) object;
    OperacaoProvisionamento o = new OperacaoProvisionamento();

    o.setEmpregado(empregado);

    retorno.add(o);
  }

  return retorno;
}
```

Tabela 11 - Código para gerar as operações de provisionamento

A procedure *sp_empregadosProvisionamento* retorna os empregados do SRH que devem ser provisionados no MS-AD. No final da tabela anterior uma lista de operações é montada com base nos empregados retornados pelo hibernate.

Depois da geração de operações de provisionamentos, as mesmas precisam ser persistidas no MS-AD, que é dado pela Tabela 12.

```
private DistinguishedName criarOUs(Empregado empregado) throws
DirSyncException {
  Directory dir = new Directory();
  dir.connect();

  String[] ous = empregado.getHierarquia().split("/");
  DistinguishedName dnOu = DomainInfo.getInstance().getDn();

  dnOu.addAtBegining(AttributeType.organizationalUnit,

  DomainInfo.getInstance().getInitialOrganizationalUnit());

  for (int i = 0; i < ous.length; i++) {
    dnOu.addAtBegining(AttributeType.organizationalUnit, ous[i]);

    OrganizationalUnit ou = new OrganizationalUnit();
    ou.setDistinguishedName(dnOu.toString());
    ou.setName(ous[i]);

    if (!dir.exists(ou)) {
      dir.create(ou);
    }
  }
```

```
  dir.disconnect();

  return dnOu;
}

private void criarUsuario(DistinguishedName dnOuBase, Empregado
empregado) throws DirSyncException {
  // Cria a dn do usuário
  dnOuBase.addAtBegining(AttributeType.commonName,
empregado.getNomeCompleto());

  // Mapeamento entre os usuários

  User usuarioAD = new User();
  usuarioAD.setDistinguishedName(dnOuBase.toString());
  usuarioAD.setGivenName(getFistName(empregado.getNomeCompleto()));
  usuarioAD.setL(empregado.getCidade());
  usuarioAD.setSAMAccountName(empregado.getLogin());
  usuarioAD.setSn(getLastName(empregado.getNomeCompleto()));
  usuarioAD.setSt(empregado.getUf());
  usuarioAD.setTelephoneNumber(empregado.getTelefone());
  usuarioAD.setUserAccountControl(UserFlags.NormalAccount.value() |
    UserFlags.AccountDisable.value() |
      UserFlags.PasswordExpired.value());
  usuarioAD.setUserPrincipalName(empregado.getLogin()+"@"+
DomainInfo.getInstance().toString());

  Directory dir = new Directory();
  dir.connect();

  dir.create(usuarioAD);

  dir.disconnect();
}
```

Tabela 12 - Código para executar as operações de provisionamento

O método executarOperacoes() utiliza os métodos auxiliares criarOUs() e criarUsuario() para realizar as tarefas do provisionamento. O método criarOUs() cria se necessário as OUs de acordo com valor do campo hieraquia do empregado. O método criarUsuario() mapeia o empregado no usuário do MS-AD e cria no diretório LDAP.

O diagrama UML da Figura 16 contém a hierarquia das classes que representam entradas no diretório do MS-AD. Essa hierarquia e os atributos foram derivados da especificação do esquema do MS-AD dado em [MSADSchema].

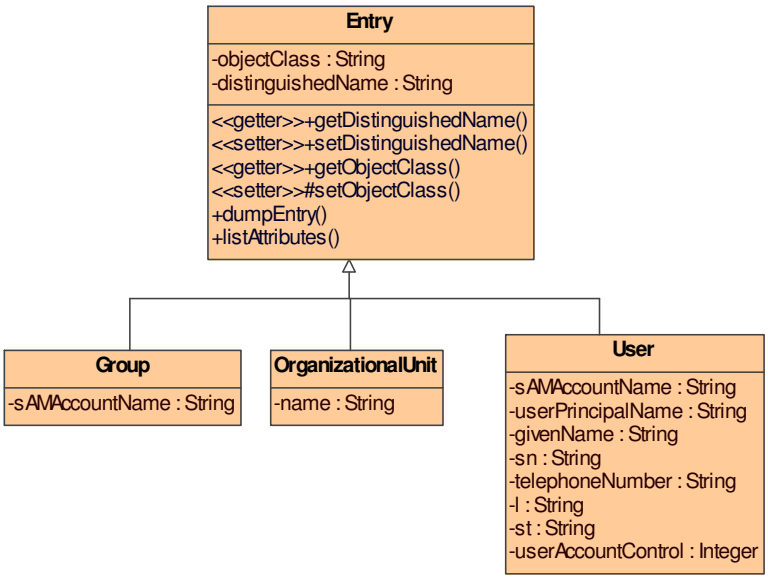

Figura 16 - Diagrama UML da hierarquia de entradas do LDAP

Note que todas as entradas do MS-AD herdam da classe Entry (que representa a classe top, com atributo adicional *distinguishedName*). Logo abaixo na hierarquia tem-se as classes Group, OrganizationalUnit e User.

O diagrama UML de algumas classes e enumerações utilitárias é dado pela Figura 17.

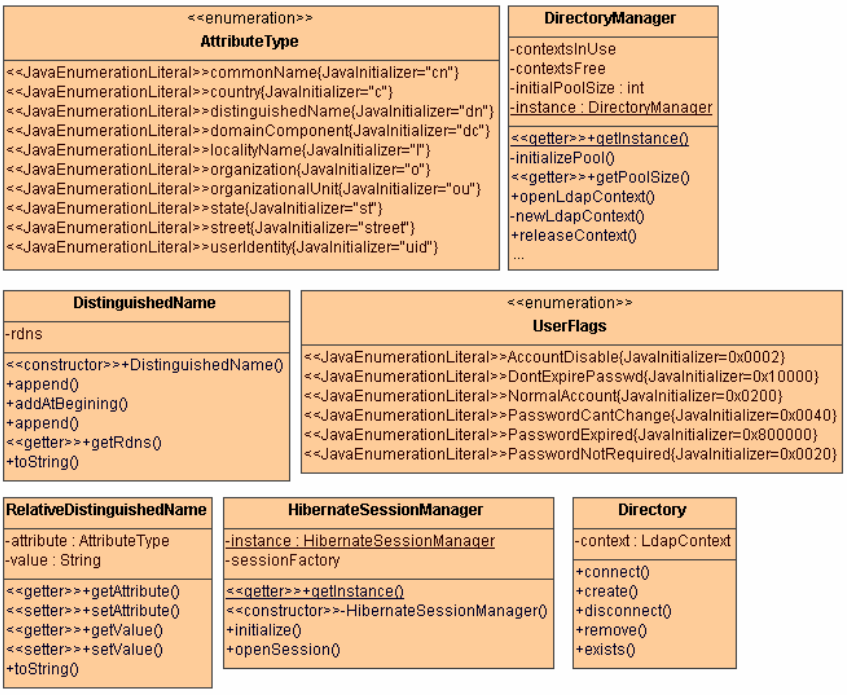

Figura 17 - Diagrama UML para classes utilitárias

A enumeração UserFlags, contém valores das flags que o atributo *userAccountControl* da classe User pode assumir. Essas flags podem ser combinadas utilizando o operador or binário conforme dito anteriormente.

As classes DistinguishedName , RelativeDistinguishedName e a enumeração AttributeType modelam o modelo de nomes do LDAP.

A classe Directory encapsula a API JNDI e contém alguns métodos para manipulação de entradas no diretório LDAP.

A classe HibernateSessionManager encapsula a API do Hibernate, por exempo contém um método para carregar as configurações o Hibernate e outro para retornar um objeto Hibernate do tipo Session.

A classe DirectoryManager implementa um pool de conexões com o diretório LDAP para aumentar o desempenho do aplicativo.

Para realização das operações restantes, respectivamente desprovisionamento, atualização e desativação um fluxo bastante semelhante ao mostrado anteriormente deve ser utilizado.

O código fonte completo desse trabalho pode ser baixado do endereço http://www.cin.ufpe.br/~sscf/dirsync

5 Conclusão e trabalhos futuros

Existem várias formas de integração do MS-AD com repositórios de dados externos, cada um com sua peculiaridade. Foi vista um conjunto de soluções que tinham como objetivo centralizar o MS-AD como repositório de dados principal.

Foi apresentado que algumas soluções que realizam especificamente a sincronização entre o MS-AD e um outro diretório (como o NIS e o OID); outras, com abordagens mais abrangentes como o MIIS e outras reimplementavam a pilha de serviços utilizados pelos aplicativos e pelo sistema operacional (através do Centrify).

Foi visto o tradeoff dessas soluções, alguma delas são demasiadamente caras (como o MIIS) e outras simplesmente incompatíveis com alguns ambientes e pouco extensíveis (como o Centrify).

A tendência das empresas brasileiras tem sido utilizar mecanismos de sincronização de diretórios atrelada com uma solução para o provisionamento automático do MS-AD através do sistema de recursos humanos. Ao contrário do citado em [Centrify] “Deployments of these solutions are very complex, frequently requiring a RDBMS to provide the data mapping.”.Porém essa solução se mostrou bastante simples e de fácil desenvolvimento. Para uma implementação mais real do protótipo desejado, devem ser realizadas as seguintes atividades:

- Abordagem de fluxos de provisionamento mais reais como a criação da pasta do usuário (quota) automaticamente.
- Abordagem de um número de atributos maior do MS-AD como o campo escritório, departamento, email, logonscript, etc.
- Abordagem da Aba Unix Attributes: Quando instalado o Services For Unix (SFU) nas propriedades do usuário do console de gerenciamento do usuário, é criada uma nova tab para manusear atributos unix. Esses atributos são armazenados no MS-AD e poderiam ser manuseados automaticamente pelo protótipo.

- Uma abordagem de provisionamento diferente de acordo com o tipo de usuário. Por exemplo, se o usuário for diretor, automaticamente ser inserido em um grupo de segurança que indique que o mesmo é diretor.
- Criação de relatório com operações realizadas no MS-AD.
- Especificar o mapeamento entre os atributos com uma notação mais elegante como XML ou Domain Specific Languages (DSL) [DSL].
- Abordagem de mais situações funcionais. No protótipo foram abordadas 3 situações funcionais (NORMAL, DEMITIDO, FÉRIAS), numa situação real foram abordados 27 situações funcionais.

Além dessas melhorias citadas anteriormente, podem ser realizadas inúmeras outras melhorias de acordo com as necessidades de cada organização.

Referências

Os links das referências abaixo foram acessados durante Novembro de 2005 e Fevereiro de 2006.

[IBMLDAP] Understanding LDAP. Endereço: http://borg.isc.ucsb.edu/aka/Ucdir/sg244986.pdf

[LDAP] Lightweight Directory Access Protocol.

Endereço: http://en.wikipedia.org/wiki/LDAP

[MSLDAP] Understandind LDAP. Endereço: http://www.microsoft.com/windows2000/techinfo/howitworks/activedirectory/ldap.asp

[INTLDAP] Introduction to LDAP. Endereço: http://ldapman.org/articles/intro_to_ldap.html

[RFCLDAP] LDAPman RFC page. Endereço: http://www.ldapman.org/ldap_rfcs.html

[RFC2849] The LDAP Data Interchange Format (LDIF). Endereço: http://www.faqs.org/rfcs/rfc2849.html Endereço:

[RFC2251] Lightweight Directory Access Protocol (v3) http://www.faqs.org/rfcs/rfc2251.html. Endereço:

iSeries Information Center Endereço: http://publib.boulder.ibm.com/iseries/v5r2/ic2924/index.htm?info/rzahy/rzahyovrco.htm

[RFC1777] Lightweight Directory Access Protocol. Endereço: http://www.ietf.org/rfc/rfc1777.txt?number=1777

[ADCompliance] Active Directory LDAP Compliance. Endereço: http://www.microsoft.com/windowsserver2003/techinfo/overview/ldapcomp.mspx

[MIIS] Identity Integration Server 2003 Home. Endereço: http://www.microsoft.com/windowsserversystem/miis2003/evaluation/overview/default.mspx

[OracleIntegration] Overview of Integration with the Microsoft Windows Environments. Endereço: http://download-west.oracle.com/docs/cd/B10464_01/manage.904/b12118/odip_ac2.htm#137056

[OracleIntegration2] Integrating Oracle Internet Directory with Microsoft Active Directory: Import Connector. Endereço: http://www.oracle.com/technology/obe/obe_as_10g/im/ads_import/import.htm

[OracleIntegration3] Windows Integration: Configuring the Import Connector. Endereço: http://www.oracle.com/technology/products/oid/oidhtml/sec_idm_training/html_masters/basics02.htm

[NET Passport] Passport Network. Endereço: http://www.passport.net/Consumer/PrivacyPolicy.asp?PPlcid=1033

[Hibernate] Endereço: http://www.hibernate.org/

[JNDI] Java Naming and Directory Interface Endereço: http://java.sun.com/products/jndi/

[Centrify] Extend Microsoft Active Directory's identity, access and policy management services to your Unix, Linux, Java and web platforms with Centrify. Endereço: http://www.centrify.com/

[Centrify2] White Paper: Centralized Identity and Policy Management for Windows, Linux, Unix, Mac and Java with Active Directory and DirectControl.

[SFU] Windows Services for UNIX. Endereço: http://www.microsoft.com/windowsserversystem/sfu/default.mspx

[MSADSchema] Active Directory Schema. Endereço: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/adschema/adschema/active_directory_schema.asp

[OracleIntegrationPlatform]. Endereço: http://www.utexas.edu/its/unix/reference/oracledocs/v92/B10501_01/network.920/a96574/odip_int.htm

[LDAPExplained] Addison Wesley: LDAP Directories Explained: An Introduction and Analysis

[DSL] Domain Specific Languages. Endereço: http://en.wikipedia.org/wiki/Domain-specific_programming_language

[MS Sharepoint] Windows SharePoint Services. Endereço: http://www.microsoft.com/windowsserver2003/technologies/sharepoint/default.mspx

[Net Passport2] Microsoft .NET Passport Software Development Kit Bits and Documentation for Windows and Non-Windows Platforms. Endereço: http://support.microsoft.com/default.aspx?scid=kb;en-us;816418

[SFU2] Microsoft Services for UNIX Requiem. Endereço:http://stephesblog.blogs.com/my_weblog/2005/09/microsoft_servi.html

[TechTarget] TechTarget. Endereço: http://www.techtarget.com