Controle de Acesso em Bancos de Dados Geográficos

Liliana Kasumi Sasaoka

Dissertação de Mestrado

Instituto de Computação
Universidade Estadual de Campinas

Controle de Acesso em Bancos de Dados Geográficos

Liliana Kasumi Sasaoka

Junho de 2002

Banca Examinadora:

- Prof. Dra. Claudia Maria Bauzer Medeiros
  Instituto de Computação, UNICAMP (Orientadora)
- Prof. Dr. Geovane Cayres Magalhães
  Instituto de Computação, UNICAMP
- Prof. Dra. Geneviève Jomier
  Université Paris IX, França
- Prof. Dr. Hans Liesenberg
  Instituto de Computação, UNICAMP (Suplente)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Sasaoka, Liliana Kasumi

Sa78c Controle de acesso em bancos de dados geográficos/Liliana Kasumi Sasaoka. -- Campinas, [S.P. :s.n.], 2002.

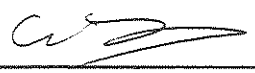
Orientador : Claudia Maria Bauzer Medeiros

Dissertação (Mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Banco de dados – Medidas de segurança. 2. Sistemas de informação geográfica. I. Medeiros, Claudia Maria Bauzer. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

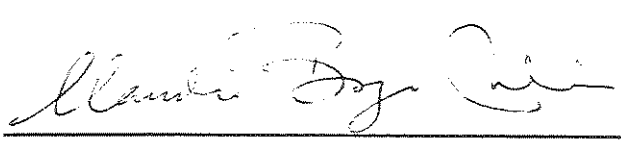
TERMO DE APROVAÇÃO

Tese defendida e aprovada em 07 de junho de 2002, pela Banca Examinadora composta pelos Professores Doutores:



Profª. Drª. Geneviève Jomier
Université Paris IX - França

Prof. Dr. Geovane Cayres Magalhães
IC - UNICAMP



Profª. Drª. Claudia Mª. Bauzer Medeiros
IC - UNICAMP

Controle de Acesso em Bancos de Dados Geográficos

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Liliana Kasumi Sasaoka e aprovada pela Banca Examinadora.

Campinas, 07 de junho de 2002.

Prof. Dra. Claudia Maria Bauzer Medeiros
Instituto de Computação, UNICAMP
(Orientadora)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Se você encontrar uma porta a sua frente, você pode abri-la ou não. Se você abrir a porta, você pode, ou não, entrar em uma nova sala. Para entrar, você vai ter que vencer a dúvida, o titubeio ou o medo. Se você venceu, você dá um grande passo: nesta sala VIVE-SE. Mas, tem um preço: são inúmeras outras portas que você descobre.

Içami Tiba

Aos meus pais e meus irmãos, com carinho.

Agradecimentos

À Deus, em primeiro lugar, pelo dom da vida e por permitir que tudo isso se tornasse realidade.

Aos meus pais, Akikazu e Mutsuko, e meus irmãos, Regina, Gerson e Edison, pelo apoio e carinho. Obrigada por toda a confiança e apoio incondicionais, que muito me ajudaram nos momentos mais difíceis.

À minha orientadora *Profª*. Claudia Bauzer Medeiros, pela compreensão, paciência e apoio. Muito obrigada pelas cobranças, sugestões e ensinamentos, que têm sido fundamentais para a minha formação profissional e pessoal.

Aos companheiros do grupo de Banco de Dados do IC, pelas inúmeras sugestões e críticas construtivas que recebi durante o desenvolvimento deste trabalho. Obrigada especialmente a Bei Yi, Daniel e Ricardo.

Aos companheiros da Fundação CPqD, pelas dicas, conselhos e sugestões. Um agradecimento especial a Hana Karina, Rogério Albertoni e Alexandre Braga, que mais diretamente colaboraram para a conclusão deste trabalho.

A todos os meus amigos e colegas, que de diversas formas me apoiaram e incentivaram. Em especial, a Renato Hirata, Heloísa Fujihara, Márcia Santos, Fabiana André Falconi e Luís Fernando Falcone Garcia.

Este trabalho foi parcialmente desenvolvido dentro do projeto SAI (Sistemas Avançados de Informação) do Núcleo de Excelência PRONEX-MCT, coordenado pelo Instituto de Computação da UNICAMP.

O desenvolvimento deste trabalho também foi parcialmente custeado com recursos do Funttel - Fundo para o Desenvolvimento Tecnológico das Telecomunicações, repassado à Fundação CPqD através da autorização da Portaria no. 581 de 08/10/2001.

Resumo

O problema de controle de acesso em bancos de dados consiste em determinar quando (e se) usuários ou aplicações podem acessar os dados armazenados, e que tipo de acesso é permitido. A maioria das soluções existentes está voltada a dados relacionais para aplicações comerciais.

O objetivo desta dissertação é estudar este problema para bancos de dados geográficos, onde as restrições impostas ao acesso são acrescidas de fatores inerentes à localização no espaço.

As principais contribuições desta pesquisa são: (a) levantamento de requisitos para controle de acesso em bancos de dados geográficos; (b) definição de um modelo de autorização baseado em caracterização espacial; (c) discussão detalhada dos aspectos de implementação deste modelo; (d) proposta de adaptação e aplicação do mecanismo para uma aplicação real na área de gerenciamento de aplicações de telefonia, o Sistema SAGRE.

Abstract

The access control problem in databases consists in determining when (and if) users or applications (WHO) can access stored data (WHAT), and what kind of access (HOW) they are allowed. Most of the research in this area is geared towards management of relational data, for commercial applications.

The objective of this thesis is to study this problem for geographic databases, where constraints imposed on access control management must consider the spatial location context.

The main contributions of this work are: (a) overview of requirement analysis for access control in geographic databases; (b) definition of an authorization model based in spatial characterization; (c) discussion of the implementation aspects of this model (d) analysis of how this proposal can be adopted by a large scale telecommunications AM/FM spatial application, the SAGRE System. SAGRE is an outside plant management geographic information system, developed at CPqD foundation, in use in most telephone operator service providers in Brazil.

Sumário

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução e motivação

Em 1979, Denning [DD79] estabeleceu uma nova definição para crime de colarinho branco. A expressão que denotava crimes onde o homem aprendera a apropriar-se de bens alheios usando uma caneta, seria estendida a partir de então para crimes envolvendo roubos usando computadores. A partir de então, os crimes digitais só têm aumentado a cada ano.

A situação agravou-se ainda mais com o advento da Internet. À medida que cresce o número de empresas que disponibilizam seus dados na Internet, aumenta também o número de crimes envolvendo conexões por Internet. A medida de segurança mais popular é o *firewall*. Um *firewall* situa-se entre a rede interna de uma empresa e a Internet. Ele monitora todo o tráfego de fora para dentro, e bloqueia qualquer tráfego que não esteja autorizado. A segurança externa de uma empresa pode ser entendida como medidas para evitar ataques de intrusos externos via Internet.

Embora *firewalls* possam proteger uma organização contra intrusos via Internet, eles são apenas defesas de primeira linha, ou seja, não são imunes a ataques internos. Uma vez que o intruso tenha obtido sucesso em entrar no sistema, os *firewalls* não provêem nenhum tipo de proteção para recursos internos. E mais ainda, os *firewalls* não protegem contra violações de segurança interna, ou seja, a partir de agentes da própria organização. Assim, segurança interna pode ser entendida como medidas para evitar ataques a partir de agentes internos a própria empresa.

De acordo com uma pesquisa realizada pela revista Information Security em 2000 [Ihr00], o número de empresas que passaram por experiências de roubos internos, isto é, roubos físicos, sabotagens ou destruições intencionais de equipamentos computacionais praticamente dobrou de 1999 para 2000. Além disso, a quantidade de empresas reportando brechas relacionadas ao controle de acesso interno aumentou em 12% durante o mesmo período.

Face a este cenário, a grande maioria das empresas tem optado por aumentar a sua

segurança externa, negligenciando a interna. Por exemplo, somente 3% de todas as empresas entrevistadas naquela pesquisa admitiram estar priorizando a segurança em bancos de dados, um elemento para aumentar a segurança interna da empresa [Ihr00]. Ihrer vai ainda mais longe e compara empresas a ovos. Os sistemas externos de segurança representam a casca, que apesar da aparência robusta não é impenetrável; os sistemas internos, como os bancos de dados, representam a gema. A maioria dos peritos em segurança acredita que os agentes internos sejam responsáveis pela vasta maioria (cerca de 80%) dos crimes envolvendo computadores [EBS95].

Como se pode ver, há uma crescente necessidade de se prover mecanismos de segurança interna. Entre eles, em especial para bancos de dados. Vários sistemas de bancos de dados comerciais vêm respondendo a esta pressão em relação à segurança, à medida que novas versões são disponibilizadas. Entre eles, podemos citar Oracle, Informix, Sybase, DB/2 e SQL-Server. Os mecanismos oferecidos são principalmente aqueles relativos à autenticação de usuários e de aplicações via palavras-chave. Há também um interesse crescente por uso de criptografia.

A segurança em bancos de dados consiste de um conjunto de medidas, políticas e mecanismos com o intuito de combinar disponibilidade à confidencialidade (*secrecy*) e integridade dos dados. Além disso, envolve também proteger o sistema de possíveis ataques executados tanto a partir de agentes internos quanto externos, maliciosos ou acidentais [BDPSN96].

Denning [Den83] lista quatro tipos de controle a fim de se obter segurança em bancos de dados: controle de acesso, controle de fluxo de informações, controle criptográfico e controle de inferências. Um controle de acesso garante que todos os acessos diretos ao sistema são autorizados de acordo com regras da política de segurança, como se pode ver na figura 1.1. O controle de acesso define quem pode acessar os objetos e termina seu papel uma vez que o acesso é concedido. O controle de fluxo de informações assegura que informações contidas em alguns objetos não fluem explicitamente (através de cópias) ou implicitamente para objetos menos protegidos que estes, e regula como a informação pode ser acessada. O controle criptográfico torna os dados incompreensíveis para qualquer pessoa exceto alguém que possua a chave para descriptografá-la. O controle de inferências protege as informações contra qualquer tipo de dedução de seu conteúdo. Esta dissertação irá ater-se somente ao controle de acesso, representado na figura 1.1.

O problema de segurança em bancos de dados está sendo agravado pelo aumento no número de usuários e tipos de aplicações. Exemplos destes novos domínios são as aplicações espaciais, CAD, biologia ou medicina. Bancos de dados espaciais são utilizados atualmente em inúmeras áreas que necessitam de um mecanismo que possa prover uma maior segurança para os dados, como aplicações médicas, militares, governamentais, ambientais, administração de recursos naturais, gerenciamento de serviços de utilidade

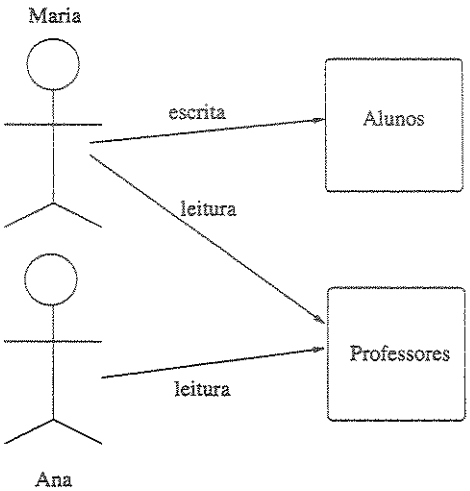


Figura 1.1: Controle de acesso.

pública, demografia e cartografia. Estes bancos de dados são a base de sistemas de informação geográfica.

Sistemas de Informação Geográfica [CCH+96], ou abreviadamente SIGs, são sistemas de informação construídos especialmente para armazenar, analisar e manipular dados geográficos, ou seja, dados que representam objetos e fenômenos em que a localização geográfica é uma característica inerente e indispensável para tratá-los. Dados geográficos são coletados a partir de diversas fontes e armazenados via de regra nos chamados **bancos de dados geográficos**.

A aceitação do SIG como uma importante ferramenta para tomada de decisões governamentais já é documentada. Até mesmo militares utilizam a tecnologia de SIG em todos os níveis do planejamento tático, operacional e estratégico, incluindo visualização e análise do terreno dos campos de batalha. Aplicações geográficas vêm motivando resultados nos vários níveis de processamento de um banco de dados na área espacial: novos tipos de dados, linguagens de consulta e sua otimização, índices e métodos de acesso [SSL99]. Entretanto, praticamente nada se encontra sobre prover um acesso seguro, organizado e controlado para dados geográficos.

Vários modelos de controle de acesso foram desenvolvidos para bancos de dados relacionais, orientados a objetos e também para alguns bancos de dados não convencionais como temporais [BBFS96b, BBFS96a, BBFS97] e de vídeo [BHAE00]. Nos últimos anos também têm surgido propostas de controle de acesso para a Web, envolvendo documentos HTML [SBJ96] e XML [DSCP01]. Estes mecanismos não podem ser diretamente usados em bancos de dados geográficos por causa das características particulares destes. O

gerenciamento de atributos com semântica associada à localização espacial (coordenadas) exige tipos distintos de controle, que passa a ser definido em termos de região geográfica. O controle de acesso pode, por exemplo, vir a exigir cortar regiões ao meio por questões como sua "visibilidade".

Como toda aplicação em banco de dados, pode haver diferentes níveis de autorização em função do perfil do usuário e do tipo de operação desejada. O que torna o controle de acesso em bancos de dados geográficos distinto é a natureza dos dados controlados. Outra questão é que pode haver diferentes níveis de granularidade de objetos espaciais em termos de acesso. Seja, por exemplo, uma aplicação que irá gerenciar a manutenção de equipamentos e infraestrutura telefônica em uma área. Nesta aplicação, deve ser possível permitir acessar apenas um objeto espacial (um poste), partes de uma região (bairro) ou ainda a região inteira (cidade de Campinas). Além da questão do objeto acessado, há o problema de quem irá acessá-lo. Ainda no exemplo, podemos citar um engenheiro da prefeitura de Campinas que esteja planejando um novo bairro e que terá um perfil de acesso distinto do perfil de um funcionário de uma empresa de telefonia que só pretende fazer reparos de equipamentos.

Um sistema específico que tem necessidade de controle de acesso geográfico é o sistema SAGRE [CCH+96]. O SAGRE (Sistema Automatizado de Gerência de Rede Externa) é um conjunto integrado de software para automatizar os diversos processos relacionados ao cadastro, planejamento, projeto, implantação, operação, manutenção, expansão e gerência da rede externa de empresas operadoras de telefonia. O SAGRE permite alcançar, dentre outros benefícios, a redução do tempo de implantação das redes telefônicas, a melhor utilização da rede instalada e a melhoria da qualidade dos serviços.

A rede externa é aquela externa às estações telefônicas e aos imóveis. Ela é composta por três tipos de redes: rede de canalização (formada por conjunto de dutos enterrados conectados a caixas subterrâneas); rede aérea (composta pelos cabos suspensos); e rede subterrânea (corresponde aos cabos que passam pela rede de canalização).

O SAGRE é um sistema construído sobre um SIG. O SIG é utilizado para gerenciar espacialmente a rede de canalização e a rede aérea sobre um mapeamento urbano. A rede subterrânea é representada em esquemáticos (ou detalhes) - *layouts*.

Como o SAGRE é utilizado em diversos setores de empresas operadoras de telefonia e pessoas com diferentes perfis têm acesso aos dados, constatou-se a necessidade de uma forma de controlar o acesso às operações que utilizam o banco de dados. Atualmente, já existe um controle de acesso baseado em perfis de usuário. O controle de acesso do SAGRE não é caracterizado pelo controle rígido aos dados armazenados a fim de preservar sigilo a qualquer preço. Ao contrário, o controle é exercido sobre funcionalidades do sistema - os usuários são autorizados por tipo de função que desejam acessar. Por isso, grande parte do foco do mecanismo de controle recai sobre o acesso a aplicações e funções por

usuários autorizados. A implementação deste mecanismo afeta fortemente a interface do SAGRE com o usuário. Cada perfil de usuário pode acessar conjuntos distintos de funções e serviços, para áreas geográficas irrestritas. Segurança, aqui, é muito mais uma questão de garantir o uso correto dos dados; protegê-los contra ataques é uma questão menor, embora, em teoria, não menos importante.

No entanto, deseja-se prover um controle de acesso levando em consideração também as informações espaciais. É óbvio que em ambientes como o SAGRE, diferentes classes de usuários dentro da mesma organização devam receber autorizações distintas para o mesmo conjunto de dados. Por exemplo, um projetista da rede externa de São Paulo não deve ter o mesmo tipo de acesso que alguém que esteja gerenciando a operação de rede na mesma cidade. Por outro lado, nenhum deveria poder atualizar dados de Campinas.

Assim, o objetivo desta dissertação é utilizar o caso SAGRE como motivação para propor um mecanismo de controle de acesso que satisfaça as necessidade de um banco de dados geográfico. A solução aqui proposta é baseada na adaptação de mecanismos de controle de acesso existentes em bancos de dados não espaciais para o contexto espacial.

O desenvolvimento da solução utilizou a seguinte linha de raciocínio. Considerado em um nível de abstração alto, o processo de autorização pode ser entendido como sendo definido a partir da seguinte seqüência de etapas:

1. Definição de regras de autorização;

2. Mapeamento destas regras em algum conjunto de estruturas de dados no banco de dados; e

3. Gerenciamento da obediência às regras.

A primeira etapa - definição das regras de autorização - pode ser especificada, em um alto nível de abstração pela frase:

> [Defina < *tipo de autorização* >
> sobre < *partição de dados no banco de dados* >]

Onde:

> *tipo de autorização*: indica classes de perfis de usuários e operações autorizadas
> *partição de dados no banco de dados*: os dados segundo o tipo de autorização

A autorização, como se verá no texto, pode ser concedida a um usuário individual ou grupos de usuários, para diferentes operações e perfis de uso/cargo na empresa. A partição do banco de dados é modelada como sendo o resultado de alguma consulta de usuário

sobre o banco de dados. O mapeamento de cada regra, por sua vez, exige determinar como armazenar uma regra de autorização de forma a garantir seu gerenciamento e obediência pelo sistema. Esta decisão de armazenamento está internamente ligada ao mecanismo que realizará tal gerenciamento e decidirá como e quando um usuário terá acesso a dados e operações.

Face a esta caracterização, o texto está organizado da seguinte forma. O Capítulo 2 apresenta uma revisão bibliográfica que introduz a terminologia e os conceitos necessários ao entendimento do texto e a literatura correlata. Os capítulos 3 e 4 tratam da etapa (1) que estuda como definir regras de autorização. O Capítulo 3 analisa as regras de acordo com usuários e operações (o *tipo de autorização*). O Capítulo 4 analisa consultas em bancos de dados geográficos visando o controle de acesso (a *definição das partições*). O Capítulo 5 aborda as etapas (2) e (3). Ele combina os conceitos dos capítulos anteriores e apresenta uma proposta de mecanismo de controle de acesso para bancos de dados geográficos. O Capítulo 6 apresenta os problemas de controle de acesso do SAGRE e discute o uso do mecanismo proposto neste contexto. E finalmente, o Capítulo 7 apresenta as conclusões finais da dissertação e suas possíveis extensões.

Capítulo 2

Conceitos básicos e revisão bibliográfica

Este capítulo introduz os conceitos básicos ao entendimento desta dissertação. A parte inicial do capítulo apresenta definições básicas (seção 2.1), seguidas de uma breve discussão sobre bancos de dados geográficos e o Sistema SAGRE (seção 2.2). A seção 2.3 apresenta uma revisão bibliográfica sobre mecanismos de controle de acesso. A seção 2.4 apresenta uma breve conclusão sobre o capítulo.

2.1 Conceitos básicos

2.1.1 Modelo de autorização

Todo mecanismo de controle de controle de acesso é baseado em algum modelo de autorização. Um modelo de autorização define como um SGBD deve implementar o controle de acesso aos dados e geralmente compreende os seguintes itens:

- granularidade de acesso;
- estruturas para representar a autorização (semântica formal de representação);
- um conjunto de políticas para gerenciar e administrar autorizações;
- algoritmos para analisar pedidos de acesso baseando-se nas autorizações.

A granularidade de acesso define a unidade de controle de acesso aos dados. Por exemplo, pode-se ter um controle de acesso em nível de tuplas, tabelas e até mesmo bancos de dados. A granularidade pode variar de acordo com o tipo do banco de dados.

Dessa forma, ela não será a mesma para bancos de dados gegráficos e bancos de dados relacionais.

Uma autorização geralmente é representada por $< s, o, m >$, onde:

- s: sujeito que recebe a autorização;
- o: objeto ao qual s terá autorização;
- m: modo de acesso (leitura, escrita, etc.).

Os objetos são entidades passivas que armazenam informações, tais como relações, tuplas em uma relação, ou mesmo elementos de uma tupla. Correspondem à definição da granularidade de acesso. Já os sujeitos são entidades ativas que acessam os objetos, podendo ser usuários, grupos de usuários ou processos que necessitam acessar os dados. O sujeito pode ainda ser um papel, onde papéis são usuários abstratos descrevendo uma posição social ou organizacional de usuário reais.

Um modelo de autorização pode, ainda, considerar a concessão de autorizações positivas e negativas. As positivas concedem o privilégio de acessar os dados e as negativas negam o acesso aos dados explicitamente.

O conjunto de políticas para administrar autorizações são regras para definir: **quem** irá conceder e revogar permissões (proprietário, administrador, qualquer usuário), **operações** (leitura, escrita), **como** estas serão executadas, de forma centralizada ou descentralizada, se haverá autorizações negativas, se haverá regras de derivação de autorizações e se houver, como estas serão derivadas.

Há várias políticas que podem ser utilizadas para administrar autorizações. Alguns exemplos são: (1) a administração centralizada, onde somente alguns usuários privilegiados podem conceder e revogar autorizações; (2) a administração baseada em propriedade, onde somente o criador do objeto pode conceder e revogar permissões; e (3) a administração descentralizada, onde usuários que não o proprietário podem conceder e revogar autorizações.

Finalmente, para que um modelo de autorização esteja completo, deve haver um mecanismo ou algoritmos que permitam validar um pedido de acesso baseado nas autorizações definidas.

Do ponto de vista de autorização, sistemas abertos são aqueles onde tudo é acessível, a menos que seja negado. Já nos sistemas fechados todas as permissões são negadas e somente aquelas que possuem uma autorização positiva é que são permitidas.

Esta dissertação utilizará como estrutura base a tupla $< s, o, m >$ e sistemas fechados. Conforme mencionado na introdução, a especificação de uma autorização será entendida como a expressão:

[Defina < *tipo de autorização* >
sobre < *partição de dados no banco de dados* >]

Mapeada para o modelo base de autorização, esta frase pode ser escrita como:

[Defina $< s, m >$
sobre $< o >$], onde $< o >$ será visto como resultado de uma consulta.

2.1.2 Controle de acesso seletivo, mandatório e baseado em papéis

Atualmente há três tipos principais de controles de acesso utilizados para prover segurança para bancos de dados [BDPSN96]: Controle de Acesso Seletivo, Controle de Acesso Mandatório e a combinação de ambos, o Controle de Acesso Baseado em Papéis.

O controle de acesso seletivo é baseado em conceder e revogar privilégios [GW76]. O acesso às informações é controlado de acordo com a identidade do usuário e com as regras que especificam os modos de acesso para cada objeto. A solicitação de um usuário para acessar um objeto é validada verificando-se a existência de uma autorização que declare que o usuário pode acessar o objeto do modo especificado. A política é dita seletiva porque os usuários podem conceder permissões de acesso aos objetos para outros usuários.

A maioria dos SGBDs comerciais utiliza o controle de acesso seletivo devido à sua flexibilidade, o que o torna apropriado para uma variedade de ambientes com diferentes requisitos de proteção. Entretanto, este modelo contém uma desvantagem: embora cada acesso seja controlado e permitido somente se autorizado, é possível burlar as restrições de acesso definidas pelas autorizações. Um sujeito que tem a permissão de ler dados pode passar os dados para outros sujeitos não autorizados sem o conhecimento do proprietário dos dados. Este ponto fraco torna o modelo seletivo vulnerável a ataques maliciosos, como Cavalos de Tróia embutidos em programas [EBS95]. Um Cavalo de Tróia é um programa com uma função útil, que contém funções adicionais ocultas que exploram clandestinamente as autorizações legítimas do processo que o invocou. Considere o seguinte exemplo para entender como um Cavalo de Tróia pode passar informações a usuários não autorizados, apesar da existência de um controle de acesso seletivo.

Suponha que Ana, uma gerente de alto nível, crie uma tabela Mercado contendo importantes informações sobre novos produtos, que deveriam ser mantidos em segredo. Considere agora Tom, um dos subordinados de Ana, que secretamente trabalha para uma outra empresa concorrente. Tom cria uma tabela Auxiliar e autoriza Ana a acessar dados desta tabela com permissão de escrita. Note que Ana nem precisa saber da existência

desta tabela. Tom então, altera uma aplicação de planilhas para incluir duas operações ocultas, uma de leitura da tabela Mercado e outra de escrita na tabela Auxiliar. Em seguida, Tom envia a nova aplicação à gerente. Suponha agora que Ana execute esta aplicação. Como resultado, informações sensíveis de Mercado são transferidos para a tabela Auxiliar e assim Tom poderá acessá-las e repassá-las ao concorrente.

O controle de acesso mandatório baseia-se em classificar hierarquicamente sujeitos e objetos do sistema, satisfazendo os requisitos de organizações militares, governamentais e comerciais que são naturalmente hierárquicas. Ele foi projetado para proteger contra invasões através de meios mais sofisticados, como Cavalos de Tróia.

A organização hierárquica classifica sujeitos e objetos em **níveis de segurança**. Típicos níveis de segurança são: altamente secreto (TS), secreto (S), confidencial (C) e não-classificado (U), onde TS>S>C>U. Existem ainda outros esquemas de classificação mais complexos, nos quais os níveis são organizadas em camadas. Entretanto, aqui serão usados apenas estes quatro níveis, a fim de facilitar o entendimento.

O modelo classifica cada **sujeito** e **objeto** em um dos níveis de segurança TS, S, C ou U e baseia-se em dois princípios formulados por Bell e LaPadula [BP76], que asseguram que a informação não flui para níveis inferiores:

- propriedade da Segurança Simples: um sujeito tem a permissão de ler um objeto somente se o seu nível de segurança é igual ou superior ao nível do objeto;
- propriedade * (Estrela): um sujeito tem a permissão de escrita sobre um objeto somente se o seu nível de segurança é igual ou inferior ao nível do objeto.

A primeira restrição é intuitiva e reforça a regra de que nenhum sujeito pode ler um objeto cujo nível de segurança seja superior ao seu. A segunda restrição é menos intuitiva. Ela proíbe um sujeito de escrever em um objeto de nível inferior, não permitindo que informações fluam de níveis superiores para inferiores.

No exemplo do Cavalo de Tróia, se Tom não tem acesso de leitura à tabela Mercado, no controle de acesso mandatório a tabela Mercado terá um nível de segurança superior ao nível de Tom. Um sujeito ou processo com permissão de leitura à tabela Mercado não seria capaz de escrever, ou seja, passar informações à tabela Auxiliar, devido a propriedade Estrela, de não poder escrever em objetos de nível inferior. Desta forma, o controle de acesso mandatório evita ataques de Cavalos de Tróia. Entretanto, pode introduzir muitas complicações quando aplicado a bancos de dados relacionais. Estas questões são discutidas na seção 2.3.3.

Existe uma vasta gama de políticas de segurança dependendo da área e das empresas onde estas são aplicadas. Cada organização, cada área possui requisitos únicos de segurança e muitas delas são difíceis de se satisfazer com apenas os tradicionais modelos de controle de acesso seletivo e mandatório.

Em muitas organizações, os usuários finais não possuem a informação a qual eles são autorizados a acessar. Para estas organizações, a corporação é a real "proprietária" dos objetos. As decisões de controle de acesso são freqüentemente determinadas pelos papéis que os usuários desempenham na empresa. Isso inclui deveres, responsabilidades e qualificações. Por exemplo, os papéis que um indivíduo pode desempenhar em um hospital incluem médico, enfermeira-chefe, enfermeira, anestesista. Uma política de controle de acesso baseada em papéis (RBAC) [FK92] tem suas decisões de acesso baseadas nas funções que um usuário pode executar dentro da organização. Os usuários não podem repassar autorizações a outros usuários de acordo com a sua vontade, sendo esta a fundamental diferença entre controle de acesso baseado em papéis e controle de acesso seletivo.

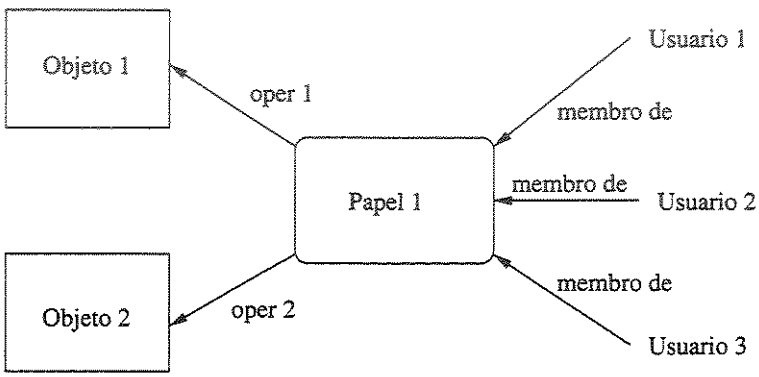


Figura 2.1: Relacionamento entre usuários, papéis, operações e objetos.

Os objetivos de segurança geralmente suportam uma política organizacional de alto nível, por exemplo como manter regras de ética. Para isso é preciso ter uma administração centralizada dos direitos de acesso. Desta forma, o administrador de segurança é responsável por fazer executar a política e ainda representar a organização.

A determinação de membros para um papel e a alocações de tarefas para um papel não é realizada de acordo com decisões arbitrárias do administrador, mas de acordo com as regras de segurança da organização. Essas regras são provenientes de leis, ética, regulamentações ou simplesmente práticas geralmente aceitas dentro da organização. Essas regras não são seletivas no sentido de que elas são impostas aos usuários. Por exemplo, um médico pode ter permissão para prescrever um remédio, mas não pode repassar esta permissão a uma enfermeira.

O controle de acesso baseado em papéis pode ser entendido como uma forma de controle de acesso mandatário, mas não é baseado em níveis de segurança. Em muitas aplicações, o controle de acesso baseado em papéis está mais preocupado em acessar

funções e informações do que estritamente com o acesso às informações. O ato de incluir um usuário em um grupo e especificar operações para determinados papéis pode ser comparado ao de conceder um nível de segurança ao usuário e aos objetos no controle de acesso mandatório. A principal preocupação do controle mandatório é: quem pode ler quais informações. Ou seja, é o fluxo de informações não autorizadas de um nível superior para um inferior. No RBAC, a principal preocupação é proteger a integridade das informações: quem pode executar quais operações sobre aquela informação. A figura 2.1 apresenta o relacionamento entre usuários, papéis, operações e objetos.

2.2 Bancos de dados geográficos e o sistema SAGRE

Os dados em bancos de dados geográficos podem ser caracterizados como tendo dois tipos de atributos:

- atributos convencionais (também chamados descritivos); e
- atributos espaciais (que referenciam os dados à superfície da Terra).

Os componentes espaciais podem ter dois tipos de formato - vetorial ou matricial. O formato vetorial permite associar a geometria dos objetos no banco de dados à sua localização. Esta geometria caracteriza objetos como sendo do tipo ponto (por exemplo, um poste), linha (por exemplo, uma rua), ou polígono (por exemplo, um lote). O formato matricial permite armazenar imagens dos dados (por exemplo, imagens de satélites).

Os componentes espaciais no Sistema SAGRE são essencialmente de natureza vetorial. A dissertação irá se limitar a estudar o controle de acesso a dados geográficos cujo componente espacial esteja em formato vetorial.

Denomina-se "rede externa" a rede que está do lado de fora das Estações Telefônicas. É o conjunto de cabos telefônicos, incluindo cabos de entrada em edifícios, fios de distribuição externa e equipamentos acessórios (excetuando-se os telefones), externos às estações telefônicas, destinados a interligar os telefones às estações, bem como estas entre si. A partir da estação devem irradiar os cabos para as outras estações e as linhas de assinantes. Estes cabos podem ser aéreos/suspensos (rede aérea), subterrâneos (rede subterrânea), em dutos ou diretamente enterrados (rede de canalização).

O SAGRE é um sistema construído sobre um SIG. O SIG é utilizado para representar espacialmente a rede de canalização e a rede aérea sobre um mapeamento urbano.

Como o SAGRE necessita de uma forma de prover um controle de acesso levando em consideração também as informações espaciais, a dissertação pretende utilizar a experiência do SAGRE e seus requisitos para definir um mecanismo adequado ao controle de acesso para bancos de dados espaciais. Os requisitos de acesso do SAGRE estão descritos no capítulo 6.

2.3 Trabalhos correlatos

Esta seção faz uma revisão de alguns trabalhos publicados na área de controle de acesso para bancos de dados.

2.3.1 Controle de acesso seletivo para o sistema R

Um dos primeiros modelos de autorização desenvolvidos como parte de um SGBD comercial foi o modelo desenvolvido por Griffiths e Wade [GW76] em 1976 no sistema R, e mais tarde revisado por Fagin [Fag78]. A granularidade de permissão do modelo considera como objetos as tabelas sobre as quais os privilégios (selecionar, inserir, atualizar e remover) são definidos. Todos os privilégios se aplicam a tabelas como um todo, exceto a permissão de atualização, que pode se referir a colunas específicas dentro de uma tabela.

O modelo provê uma administração de autorizações descentralizada, onde qualquer usuário pode criar uma tabela, tornando-se o proprietário desta com todos os privilégios possíveis. O proprietário pode conceder permissões a outros usuários também com a opção de que estes possam conceder permissões a outros. Entretanto, somente o proprietário de uma tabela pode removê-la. Uma autorização pode ser modelada como uma tupla $< s, p, t, ts, g, go >$, onde:

- s: sujeito;
- p: modo de acesso, podendo ser selecionar, inserir, atualizar e remover;
- t: tabela;
- g: usuário que concede a autorização;
- ts: instante em que a autorização foi concedida;
- go: se go = "sim", o sujeito tem a opção de conceder permissões a outros usuários.

A revogação de uma permissão de um usuário por outro é executada considerando-se como válidas somente as que teriam resultado se o revogador nunca tivesse concedido a permissão. Como conseqüência, toda vez que uma permissão é revogada, uma revogação recursiva (revogação em cascata) é executada.

As visões são utilizadas como um importante mecanismo para este modelo de autorização. Por exemplo, se o usuário A da relação R deseja que o usuário B possa ler somente alguns campos de R, então A pode criar uma visão a partir de R que inclua somente os atributos desejados, e então conceder a permissão de leitura ao usuário B somente para esta visão. O mesmo pode ser aplicado para especificar que B possa acessar somente algumas tuplas de R.

2.3.2 Extensões ao modelo do sistema R

A fim de simplificar o gerenciamento de autorizações, Wilms e Lindsay [WL82] estenderam o modelo de autorização do sistema R incluindo autorizações para grupos de usuários. Grupos podem conter usuários ou mesmo outros grupos e não precisam ser disjuntos, isto é, um usuário ou grupo pode pertencer a mais de um grupo.

Em [EBS97] Bertino e outros propuseram uma outra extensão para bancos de dados relacionais usando autorização negativa e revogação não-cascata de permissões. O sistema R utiliza a política de sistemas fechados, onde a inexistência de uma autorização é interpretada como uma autorização negativa. Dessa forma, toda vez que um sujeito tenta acessar dados e uma autorização positiva não é encontrada, seu acesso é negado. Entretanto isso não garante que este sujeito não irá acessar os dados mais tarde, já que qualquer outra pessoa que tenha acesso a esses dados pode lhe conceder o acesso. O uso de autorizações negativas soluciona este problema; e conflitos existentes entre autorizações positivas e negativas são resolvidos dando-se precedência às negativas.

A revogação não-cascata é aquela que não revoga autorizações recursivamente. Assim, autorizações concedidas pelo sujeito B que está tendo as permissões revogadas passam a ser definidas como se tivessem sido concedidas pelo sujeito A, que está revogando os privilégios do sujeito B. Este tipo de revogação é uma vantagem para organizações onde as autorizações de um usuário estão relacionadas ao seu cargo na organização. Se um usuário muda de cargo, por exemplo se for promovido, é desejável remover somente as permissões deste usuário, e não de todos os outros a quem este concedeu permissões.

Em [EBS96] Bertino e outros propuseram ainda uma outra extensão, usando um mecanismo de autorização mais flexível que suporta múltiplas políticas de controle de acesso, a política do sistema aberto e a do sistema fechado. Isso resolve os problemas para os casos em que os requisitos de controle de acesso de uma aplicação são diferentes da política implementada pelo mecanismo. A separação entre política de segurança e mecanismo de segurança é uma grande vantagem no sentido de poder alterar a política sem ter que alterar o mecanismo.

2.3.3 Controle de acesso mandatório para bancos de dados relacionais

A implementação do controle de acesso mandatório em bancos de dados relacionais resulta em bancos de dados multiníveis [EN97], onde as relações passam a conter vários níveis e não somente um, como no modelo tradicional.

Para incorporar a noção de segurança multinível ao modelo de dados relacional, é necessário que todos os dados armazenados na relação sejam classificados em níveis de segurança TC<S<C<U. É comum considerar valores de atributos e tuplas como objetos.

Portanto, cada atributo A é associado a uma classificação de atributos C no esquema; e cada valor de atributo em uma tupla é associada com a classificação correspondente. Além disso, alguns modelos adicionam uma classificação de tupla TC à relação a fim de prover uma classificação para cada tupla como um todo.

A associação de níveis de segurança aos dados de uma relação introduz a noção de relação multinível. Uma relação multinível com n atributos pode ser representada como:

$$R(A_1, C_1, A_2, C_2, ..., A_n, C_n, TC)$$

onde cada C_i representa a classificação associada ao atributo A_i.

O valor do atributo TC (que é o maior nível de segurança dos valores dos atributos da tupla t) em cada tupla t provê uma classificação genérica para a tupla. Cada C_i provê uma classificação mais detalhada para cada valor de atributo em uma tupla.

Em uma relação multinível existe o conceito de **chave aparente**. Chave aparente é um conjunto de atributos que formariam a chave primária em uma relação convencional. Uma relação multinível deve apresentar diferentes dados para sujeitos com diferentes níveis de segurança. Em alguns casos, é possível armazenar uma única tupla na relação em um determinado nível e produzir uma tupla correspondente em um nível inferior usando um processo chamado **filtragem**. Já em outros casos, é necessário armazenar duas ou mais tuplas em diferentes níveis com os mesmos valores para a chave aparente. Isto é chamado de **poliinstanciação**. Neste caso, várias tuplas têm a mesma chave aparente e diferentes valores de atributos para níveis de segurança distintos.

Estes conceitos estão ilustrados nas tabelas 2.1, 2.2, 2.3 e 2.4. Cada classificação de atributo é apresentada ao lado do valor do atributo. Suponha que o atributo Nome seja a chave aparente, e considere a consulta *SELECT * FROM empregado*. Um sujeito com um nível S pode visualizar somente as tuplas da relação mostrada na tabela 2.1, desde que os níveis das tuplas sejam menores ou iguais a S. Entretanto, um sujeito com nível C não pode visualizar dados do Salário de João, nem o Desempenho de José, pois estes têm uma classificação superior. As tuplas podem ser filtradas e mostradas como na tabela 2.2, com os valores para Salário e Desempenho nulos. Para um sujeito com nível U, o processo de filtragem permite apresentar apenas o nome de José, com todos os outros valores nulos (tabela 2.3). O processo de filtragem mascara os valores reais introduzindo valores nulos para valores de atributos com classificação superior ao nível do sujeito.

Nome	Salário	Desempenho	TC
José U	2.000 C	razoável S	S
João C	4.000 S	bom C	S

Tabela 2.1: Tabela Empregado original.

Nome	Salário	Desempenho	TC
José U	2.000 C	nulo C	C
João C	nulo C	bom C	C

Tabela 2.2: Tabela Empregado após filtragem para sujeitos do nível C.

Nome	Salário	Desempenho	TC
José U	nulo U	nulo U	U

Tabela 2.3: Tabela Empregado após filtragem para sujeitos do nível U.

Nome	Salário	Desempenho	TC
José U	2.000 C	razoável S	S
José U	2.000 C	excelente C	C
João C	4.000 S	bom C	S

Tabela 2.4: Tabela Empregado poliinstanciada para a tupla de José .

A regra de integridade de entidades para relações multiníveis define que todos os atributos que fazem parte da chave aparente não devem ser nulos e devem ter o mesmo nível de segurança em cada tupla. Em contrapartida, todos os outros valores de atributos devem ter um nível maior ou igual ao da chave aparente. Esta restrição assegura que um sujeito pode sempre visualizar a chave, caso tenha permissão de ler qualquer outra parte da tupla.

Para ilustrar melhor o conceito de poliinstanciação, suponha que um sujeito de nível C tente atualizar o Desempenho de José (vide tabela 2.2) para 'excelente'. Isto corresponde ao seguinte comando *SQL*: *UPDATE empregado SET desempenho = 'excelente' WHERE nome = 'José'*.

O sistema deve permitir tal comando de atualização, caso contrário, o sujeito pode inferir que algum valor não nulo já exista para o atributo Desempenho de José. Este tipo de inferência de informações é conhecido como canal velado (*covert channel*), ou seja, o sujeito de uma classe inferior não deve conseguir distingüir entre valores nulos reais e valores nulos resultantes de mascarações. Por outro lado, o sujeito também não pode sobrescrever esse dado de um nível superior. A solução é criar uma poliinstanciação para a tupla de José com o nível C, como se pode ver na tabela 2.4. Isto é necessário porque a nova tupla não pode ser filtrada a partir da tupla existente de nível S.

O exemplo da tabela 2.4 apresentou apenas a poliinstanciação de atributos, mas ela pode afetar também relações e atributos. Temos então os seguintes tipos de poliinstanciação:

- relações poliinstanciadas: são relações identificadas pelo mesmo nome, porém cujos esquemas têm diferentes níveis de segurança;
- tuplas poliinstanciadas: são tuplas com a mesma chave aparente, mas com diferentes níveis de segurança associadas à chave;
- atributos poliinstanciados: são valores de um atributo, que têm diferentes níveis de segurança e estão associados à uma mesma chave que possui o mesmo nível de segurança.

A poliinstanciação de tuplas e atributos ocorre em duas situações:

- quando um usuário de um determinado nível insere dados em um atributo que já contém dados em um nível superior. Isso é conhecido como poliinstanciação invisível, porque o usuário do nível inferior não sabe que está causando uma poliinstanciação;
- quando um usuário de nível superior insere dados em um campo que já contém dados em um nível inferior. Nesse caso, a poliinstanciação é dita visível.

Várias técnicas têm sido propostas para tentar solucionar este problema, mas cada uma tem vantagens e desvantagens. Algumas optam por não permitir a poliinstanciação. Entretando, isso restringe a flexibilidade do sistema. As que optam por permiti-la, baseiam-se no fato de que a poliinstanciação é um fato inevitável para o modelo multinível. Usuários de diferentes níveis de segurança devem ver diferentes valores para uma mesma tupla do mundo real. Isso também é válido para prover dados falsos, a fim de que usuários não possam inferir dados existentes em classes superiores.

O problema de inferência de dados em bancos de dados multiníveis é tratado nos trabalhos de Marks [Mar96] e Delugach e Hinke [DH96]. Marks propõe novas técnicas para

lidar com isso, derivando condições suficientes para detectar possibilidades de inferências proibidas. Delugach e Hinke descrevem o sistema de análise e detecção de inferências chamado *Wizard*, que permite detectar problemas de inferência no projeto do banco de dados, analisando os esquemas, instâncias (se estiverem disponíveis) e informações do domínio fornecidas por projetistas.

Bancos de dados multiníveis requerem que as operações básicas de atualização do modelo relacional (inserção, remoção e atualização) sejam modificadas para suportar as situações de poliinstanciação.

Um dos grandes problemas existentes em bancos de dados multiníveis é o processamento de transações. Em [EBS95] os autores ressaltam que os dois mecanismos de controle de concorrência mais conhecidos, *two-phase locking* e ordenação *timestamp*, não satisfazem os requisitos de segurança, pois ambos podem causar canais de sinalização (*signaling channels*), transmitindo sinais de transações de níveis superiores a transações de níveis inferiores quando ambas estiverem sendo executadas simultaneamente. O grande objetivo do processamento de transações seguras é garantir não só a consistência mas também a segurança do sistema. Uma alternativa para resolver o problema é manter múltiplas versões de dados ao invés de uma única versão. Usando-se esta alternativa, elimina-se a possibilidade de canais de sinalização.

Uma outra alternativa foi proposta por Bertino e Jajodia [EBM98] estendendo o mecanismo de *two-phase locking* para sistemas de arquivos multiníveis seguros. Este modelo baseia-se em bloqueios (*locks*) de dados com uma única versão. Seu intuito é prover um controle de concorrência seguro, porém sem se preocupar com o desempenho.

2.3.4 Controle de acesso para bancos de dados orientados a objetos

A maior parte dos trabalhos envolvendo o controle de acesso mandatório propõe modelos de autorização para bancos de dados relacionais [KPSN96, AJ96, ST90, QL96], tentando ao mesmo tempo resolver os problemas de conflitos entre segurança e integridade. Entretanto, recentemente (em 1996), um estudo envolvendo o controle de acesso mandatório e bancos de dados orientados a objetos foi apresentado [TS96], propondo uma arquitetura para projetar um banco de dados orientado a objetos multinível seguro.

A tecnologia de orientação a objetos tem tido larga utilização por cobrir semânticas do mundo real de modo mais explícito, atendendo assim necessidades de aplicações que requerem estruturas de dados mais complexas do que apenas relações. Todavia, enquanto novas tecnologias trazem facilidades, essas mesmas facilidades requerem considerações para se evitar a introdução de novos problemas. A rica semântica do modelo de dados OO introduz novos requisitos para o controle de acesso, que os mecanismos tradicionais

não provêem.

Em primeiro lugar, o modelo de autorização deve levar em consideração todos os relacionamentos semânticos que possam existir entre os dados (herança, composição). Em segundo lugar para executar alguma operação em um dado objeto (instância), o usuário deve ter autorização para acessar outros objetos, como por exemplo a classe à qual a instância pertence ou objetos que são componentes deste objeto. Finalmente, a administração de autorizações torna-se muito mais complexa. Em particular, o conceito de proprietário não tem uma interpretação clara no contexto OO. Por exemplo, um usuário pode criar uma instância de uma classe que não lhe pertence devido à composição de objetos. Neste caso, quem seria considerado o proprietário da instância? Diferentes níveis de granularidade precisam ser suportados. Em sistemas de bancos de dados OO, os objetos é que são as unidades de acesso. Portanto, os mecanismos de autorização devem permitir associar autorizações a objetos. Por outro lado, tal granularidade pode comprometer o desempenho quando estiver acessando conjuntos de objetos, como no caso de consultas. Assim, o mecanismo de autorização deve permitir que usuários associem autorizações a classes ou mesmo a hierarquias inteiras.

Alguns destes problemas foram tratados nos trabalhos [EBFS94, RBKW91]. Kim e seu grupo [RBKW91] desenvolveram um modelo de autorização para o protótipo Orion usando o conceito de autorização implícita no domínio de sujeitos, objetos e modos de acesso. Por exemplo, uma permissão de leitura em uma classe implica em permissões de leitura em todas as instâncias da classe. Regras para sujeitos dependem dos relacionamentos especificados entre papéis de usuários. Por exemplo, uma ligação entre secretária e empregada indica que todas as secretárias são também empregadas; portanto, todas as autorizações para empregadas são válidas também para secretárias.

Fernandez [EBFS94] propõe um modelo usando o conceito de autorização implícita ao longo de uma estrutura hierárquica de classes. Assim, um usuário que tem permissão de leitura para uma classe também tem permissão para todas as subclasses desta classe, restrita entretanto somente aos atributos herdados da classe pai.

Esses dois modelos consideram apenas operações de leitura e escrita e não exploram a propriedade de encapsulamento da técnica OO. Encapsulamento significa que objetos podem ser acessados somente através da invocação de métodos pré-definidos. Bertino [Ber92] propõe um modelo onde autorizações especificam privilégios para usuários executarem métodos sobre objetos. Também usa o conceito de método privado para permitir a execução controlada de alguns métodos.

Todos estes trabalhos apresentados utilizam a política de controle de acesso seletivo. Há ainda, algumas propostas que tentam combinar a política de controle de acesso mandatório e bancos de dados OO [NO, TS96].

O modelo de Bell-LaPadula [BP76] é baseado no paradigma sujeito-objeto e a aplicação

deste paradigma para sistemas OO não é direta, além de ser considerada como sendo um pouco forçada para sistemas OO. O problema é que a noção de objeto em OO não corresponde à noção de objeto do modelo de Bell-LaPadula. O paradigma OO combina as propriedades de repositório passivo (representado por atributos e seus valores) com propriedades da entidade ativa, representadas pelos métodos e suas chamadas. Portanto, o objeto OO pode ser considerado com uma fusão do objeto e do sujeito de Bell-LaPadula. Um outro problema é associar classificações às informações armazenadas nos objetos devido aos relacionamentos semânticos entre objetos. Por exemplo, a classe de acesso de uma instância não pode ser inferior à classe de acesso da classe que a contém, caso contrário um usuário jamais conseguiria acessá-la.

Um ponto comum das propostas apresentadas para o controle de acesso em sistemas OO é que elas requerem que objetos tenham um único nível, isto é, todos os atributos de um objeto devem pertencer à mesma classe de acesso. Isso tem a vantagem de tornar a verificação da validade da autorização mais simples. No entanto, entidades do mundo real são multiníveis, alguns atributos podem ter diferentes níveis de acesso. Essa abordagem acaba restringindo a flexibilidade do sistema.

2.3.5 Controle de acesso temporal para bancos de dados

Um outro tipo de problema de acesso diz respeito a dados, em que permissões podem ser válidas apenas por um determinado período de tempo. Como os típicos SGBDs comerciais não provêem mecanismos para autorização temporal, a única solução acaba sendo implementar um gerenciador de autorizações no nível da aplicação. No entanto, esta solução é bastante inadequada, pois aumenta a complexidade da especificação e do gerenciamento de autorizações.

Várias questões devem ser consideradas em um modelo de autorização temporal: a definição de uma semântica formal para o modelo, o desenvolvimento de estratégias para um controle de acesso eficiente e ferramentas para administrar as autorizações. Nesta área destaca-se particularmente um grupo de quatro pesquisadores: Bertino, Bettini, Ferrari e Samarati, que propuseram três modelos para controle de acesso temporal em bancos de dados [BBFS96b, BBFS97, BBFS96a].

O primeiro [BBFS96b] apresenta um modelo de controle de acesso seletivo, onde as autorizações contêm intervalos de tempo de validade. Uma autorização é automaticamente revogada quando este intervalo de tempo expira. O modelo ainda provê a especificação de regras de derivação automática de novas autorizações, expressando dependências temporais entre autorizações. Por exemplo, uma regra de derivação pode definir que um usuário A pode ler um objeto contanto que um outro usuário B possa lê-lo também. As regras de derivação permitem derivar novas autorizações baseadas na presença ou ausência de

outras autorizações. Tanto autorizações positivas quanto negativas são suportadas e, em caso de conflitos, utiliza-se a regra de precedência das autorizações negativas. Além disso, o modelo apresenta ainda estratégias baseadas em técnicas de materialização de visões a fim de suportar uma verificação eficiente de autorizações.

Neste modelo uma autorização pode ser definida como uma tupla da forma <tempo, autorização> ou melhor $< (ti, tf), (s, o, m, pn, g) >$, onde:

- ti: tempo inicial;
- tf: tempo final;
- s: usuário que recebe a autorização;
- o: objeto;
- m: modo de acesso;
- pn: se $pn = +$, s recebe a autorização, se $pn = -$, a autorização é negada;
- g: usuário que concede a autorização.

Este primeiro modelo, porém, não provê autorizações periódicas, o que, segundo Bertino [EBS95], é um item essencial para um mecanismo de controle de acesso temporal. Em algumas organizações, as autorizações concedidas a usuários devem ser feitas sob medida para o padrão de atividades de cada usuário dentro da organização. Portanto, usuários devem ter autorizações de acesso somente para o período de tempo no qual se espera que eles necessitem dos dados. Um exemplo de autorização é "um estagiário de tempo parcial deve ter acesso aos dados todos os dias úteis das 8:00 às 12:00". Além disso, autorizações periódicas também são muito importantes quando se tratam de autorizações de execução para aplicações. O controle do período de tempo em que uma determinada aplicação pode ser invocada é útil para a otimização da utilização dos recursos. Programas, cuja execução necessita de muitos recursos, poderiam ser alocados para períodos nos quais outros programas não fossem executados. Entretanto, autorizações periódicas são mais complexas de se manipular do que as não periódicas e a solução de implementá-las como parte de uma aplicação também torna-se inviável.

Assim, mais tarde, o mesmo grupo propôs uma extensão ao seu modelo de controle de acesso provendo autorizações e regras para acesso periódico [BBFS96a], permitindo que autorizações pudessem ser periódicas e ter um tempo limitado de validade. Dessa maneira, uma autorização é automaticamente concedida no tempo especificado por uma expressão periódica (definida pelo modelo) e revogada quando esta expira. Assim como no modelo anterior, novas regras são derivadas a partir das já existentes. A diferença

nesta abordagem é que as novas regras são baseadas na presença ou ausência de outras autorizações em períodos específicos de tempo. A estratégia de materialização também foi substancialmente estendida para atender às novas necessidades de periodicidade e restrições de ordem.

Neste modelo estendido uma autorização periódica é expressa como uma tupla da forma <tempo, período, autorização>, onde:

- tempo: intervalo de tempo [min, max], min e max denotam os instantes de tempo inicial ti e final tf com $0 \leq ti \leq tf$;
- período: uma expressão periódica;
- autorização: autorização da forma $< s, o, m, pn, g >$.

A expressão periódica pode ser o conjunto das segundas-feiras, os grupos das terceiras horas dos primeiros dias de cada mês e assim por diante, e elas são representadas formalmente por uma expressão definida como P. Portanto, $< [min, max], P, (s, o, m, pn, g) >$ denota que o usuário g concedeu uma autorização com privilégio m ao usuário s para acessar o objeto o, que é válida para cada instante em P limitado pelo intervalo min e max.

O primeiro modelo apresentado em [BBFS96b] considera privilégios administrativos especiais que permitem usuários concederem e revogarem autorizações. Por exemplo, um usuário com privilégios de administrador sobre um objeto pode conceder e revogar autorizações de qualquer modo de acesso sobre este objeto.

O terceiro modelo [BBFS97] é uma extensão do primeiro modelo, considerando desta vez uma administração descentralizada para o modelo de controle de acesso temporal. A política administrativa é estendida para permitir que o proprietário do objeto e usuários com o privilégio de administrador possam conceder e revogar permissões de qualquer tipo sobre o objeto. Além de poder delegar privilégios de administrador sobre o objeto, o proprietário ainda pode seletivamente conceder autorizações com a permissão de *grant*, o que não é permitido no primeiro módulo. Isso permite delegar administrações de um tipo de privilégio (leitura, escrita) para determinados intervalos de tempo. Embora isso possa causar uma certa perda de controle, o proprietário pode ainda deter algum controle sobre seu objeto, definindo explicitamente uma autorização negativa a um determinado usuário. Portanto, este modelo provê um alto grau de flexibilidade à medida em que suporta uma administração descentralizada. A revogação de privilégios é recursiva. Como cada autorização tem um intervalo de tempo associado, uma solicitação de revogação pode causar não só a remoção de autorizações, onde a remoção é explicitamente solicitada, como também a alteração dos intervalos de tempo ou a divisão de uma autorização em várias outras com intervalos de tempo disjuntos.

2.3.6 Controle de acesso para bancos de dados de vídeo

Modelos de autorização em geral são definidos de forma genérica. Existem alguns trabalhos que tratam de autorização para situações especiais. Um exemplo é o controle de acesso para bancos de dados de vídeo. O que o torna diferente é que os acessos aos vídeos são geralmente descritos em termos de seu conteúdo semântico. Por exemplo, pode-se conceber que filmes violentos sejam proibidos para menores de 10 anos.

Um ponto importante neste problema é a necessidade de modelos e mecanismos que suportem a especificação de autorizações baseados nas características, tarefas, qualificações e cargos dos usuários, ao invés de utilizar somente a sua identidade. Um outro ponto é a necessidade de uma ferramenta que suporte autorizações baseadas em conteúdo. O terceiro ponto é o fato de que os dados de vídeo têm uma estrutura hierárquica: um vídeo inteiro, seqüências de telas (*frames*) de um vídeo e objetos de um vídeo. Isso causa a necessidade de um mecanismo que suporte várias granularidades.

Um exemplo de controle de acesso para bancos de dados de vídeo é o modelo proposto em [BHAE00], que permite controlar o acesso baseado no seu conteúdo semântico e nas credenciais do usuário, o que é uma forma mais flexível e natural de expressar autorizações para dados de vídeo. Assim, diferentes pessoas podem assistir a diferentes visões do mesmo vídeo, de acordo com o seu perfil.

No modelo, os dados do vídeo são modelados por três elementos: um vídeo completo (*video stream*), um segmento de vídeo e um objeto de vídeo. Um vídeo completo representa a seqüência contínua de telas (*frames*) que formam o contexto do vídeo. Um segmento de vídeo é uma seqüência de telas interrelacionadas. Tanto o vídeo completo quanto o segmento de vídeo estão associados com anotações textuais que os descrevem semanticamente. Já o objeto de vídeo representa um objeto visual como uma face, um carro ou um prédio. Cada objeto de vídeo tem várias ocorrências e cada ocorrência tem suas características espaço-temporais que descrevem sua geometria, seqüência de polígonos ou mínimos retângulos envolventes (MBRs - *minimum bounding rectangles*) representando os limites do objeto durante um certo período de tempo. Tanto o objeto quanto a sua ocorrência possuem anotações textuais associadas. Todas estas informações textuais é que permitem suportar autorizações baseadas em conteúdo usando uma definição formal proposta no modelo, por exemplo: x.anotação contém "violência".

Os sujeitos são usuários finais com um conjunto de informações sobre suas características. Essas informações utilizadas em conjunto com uma expressão de perfil é que possibilitam o acesso baseado no perfil do usuário. Um exemplo de expressão de perfil seria: x.idade > 18.

Objetos podem ser um elemento de vídeo ou um conjunto de elementos de vídeo. Um elemento de vídeo pode ter partes censuradas, a fim de restringir o acesso. Neste caso, quando alguém sem acesso ao elemento de vídeo tenta ver o vídeo, pode ter partes da

imagem borradas. Quando as partes censuradas são telas inteiras pode-se por exemplo realizar um processo de filtragem.

Como operações como leitura e escrita não fazem muito sentido para o controle de acesso em vídeos, um novo conjunto de operações abstratas é introduzido: visualizar anotações, visualizar telas, executar um vídeo (*play*) por um período e com uma certa qualidade de imagem, editar anotações e editar vídeos (modificar ou adicionar elementos de vídeo a um banco de dados de vídeo).

2.4 Conclusão

Este capítulo apresentou os principais conceitos relativos ao controle de acesso e trabalhos correlatos. Todo mecanismo de controle de acesso tem como base um modelo que define como deve ser a concessão de autorizações. Este modelo obrigatoriamente define os objetos a serem acessados, os tipos de acesso permitidos a estes objetos e quem/o que pode acessá-los.

Além dos trabalhos apresentados, há outros que se ocupam de situações onde há vários bancos de dados em "federação" [JD94, IGC94]. Uma federação de bancos de dados integra SGBDs existentes preservando a sua autonomia. Os seus componentes (SGBDs) não precisam ser homogêneos, ou seja não precisam ser todos relacionais). Entretanto, esses trabalhos não são aplicáveis ao problema abordado.

Esta dissertação propõe um mecanismo de controle de acesso para aplicações geográficas. Neste sentido, aproxima-se do trabalho de Bertino [BHAE00] de aplicações de vídeo, pois não é um mecanismo genérico e sim voltado a sistemas com uma semântica específica. O acesso pode ser definido por valores de atributos como em mecanismos genéricos, mas também usando semântica espacial. Pelo fato de dados geográficos serem modeláveis como objetos complexos, os problemas a serem considerados também envolvem questões relativas a acesso em bancos de dados orientados a objetos.

Os próximos dois capítulos caracterizam o mecanismo proposto nesta dissertação segundo o modelo $< s, o, m >$.

Capítulo 3

Usuários e operações

O objetivo do controle de acesso é limitar as ações ou operações que um usuário legítimo do sistema pode executar sobre o banco de dados. O controle de acesso restringe o que um usuário pode fazer diretamente, bem como programas executando sob o seu comando.

A fim de controlar o acesso, usuários recebem autorizações para executar determinadas operações sobre partes do banco de dados. Como visto no capítulo 2, a estrutura adotada para representar uma autorização é baseada em uma tupla $< s, o, m >$, onde s é o sujeito que recebe a autorização, o o objeto ao qual se terá autorização e m o modo de acesso, ou seja, a operação a qual o usuário pode executar.

Este capítulo trata dos sujeitos e dos tipos de operações que estes podem executar sobre os dados. A seção 3.1 apresenta os sujeitos das autorizações e a seção 3.2 apresenta os tipos de operações. Finalmente, a seção 3.3 apresenta as conclusões do capítulo.

3.1 Sujeitos

Sujeito é a pessoa que recebe a autorização. O controle de acesso para bancos de dados geográficos não apresenta diferenças quanto ao tipo de usuário, isto é, os usuários podem ser definidos da mesma forma que nos bancos de dados convencionais.

Um sujeito pode representar uma pessoa, uma conta de usuário, um grupo de usuários, ou mesmo processos ou programas executando sob o comando de um usuário. O sujeito pode ainda representar determinados papéis ou perfis dentro de uma empresa, já que a autorização pode estar associada muitas vezes não ao usuário, mas sim ao cargo que este ocupa na empresa. Assim, cada papel pode ter a ele associado um conjunto específico de permissões de acordo com o cargo representado. Por exemplo, pode-se determinar que o perfil de "diretor de recursos humanos" tenha determinadas autorizações para modificar salários de outros funcionários. Já o perfil "secretária" pode apenas visualizar os salários dos funcionários. Este é o controle de acesso baseado em papéis (vide seção 2.1.2).

Existe ainda um tipo especial de sujeito que possui determinados privilégios sobre o banco de dados [EN97], justamente para poder gerenciá-lo e também conceder autorizações ao sujeitos comuns. Este sujeito é o administrador do banco de dados, mais conhecido como DBA (*DataBase Administrator*) e é a autoridade central que gerencia um banco de dados.

As responsabilidades do DBA incluem conceder privilégios a usuários que necessitam utilizar o sistema e classificá-los de acordo com a política da empresa. O DBA possui uma conta de administrador no banco de dados, muitas vezes chamada de *system* ou conta de superusuário. Essa conta provê funções que não são disponibilizadas a usuários ou contas comuns do banco de dados. Alguns dos privilégios do DBA incluem comandos para conceder e revogar autorizações de outras contas, usuários ou grupos de usuários. Também fazem parte de suas atividades:

1. Criação de conta: cria uma nova conta e senha para um usuário ou grupo de usuários a fim de permitir que possam acessar o banco de dados.

2. Concessão de privilégio: concede determinados tipos de privilégios a determinadas contas.

3. Revogação de privilégio: revoga (cancela) determinados privilégios que haviam sido previamente concedidos a determinadas contas.

4. Associação de níveis de segurança: associa níveis de segurança apropriados a cada conta de usuário.

O DBA é responsável pela segurança do sistema de banco de dados como um todo. A atividade (1) lhe permite controlar o acesso a todo o banco de dados. Já as atividades (2) e (3) são usadas para controlar o acesso seletivo e a atividade (4) para controlar o acesso mandatório.

Finalmente, sujeitos podem também ser definidos como módulos de software. Neste caso, a autorização determina que módulos podem acessar os dados. Esta dissertação ignorará este tipo de sujeito, partindo do pressuposto que o controle de acesso via software deve ser gerenciado por mecanismos distintos daqueles tratados por SGBD.

3.2 Operações

O m da tupla $< s, o, m >$ corresponde ao modo de acesso, ou seja, o tipo de operação que o sujeito tem permissão de executar sobre o dado. O modelo de Griffiths e Wade [GW76] para bancos de dados relacionais definia que o sujeito poderia executar os seguintes tipos de operações sobre tabelas do banco de dados:

- Seleção (*select*), seleção de tuplas de uma tabela;
- Inserção (*insert*), inserção de tuplas em uma tabela;
- Remoção (*delete*), remoção de tuplas de uma tabela;
- Atualização (*update*), atualização de tuplas em uma tabela.

Neste modelo, todas as operações se aplicavam à tabela como um todo, exceto a atualização, que poderia se referir a colunas específicas de uma tabela. Esse modelo também suportava uma administração descentralizada de autorizações, que permitia que qualquer usuário criasse uma tabela, tornando-se o seu proprietário. Um proprietário tinha todas as permissões sobre a sua tabela, inclusive uma permissão especial de concessão (*grant*) para conceder autorizações a outros usuários sobre a sua tabela, inclusive podendo repassar autorização de concessão (*grant*). Todo usuário que concedia uma autorização a outro, também tinha a permissão de revogação (*revoke*) desta autorização.

Em 1996, Dastjerdi [BDPSN96] definiu o conjunto básico de operações como: leitura (*read*), escrita (*write*), remoção (*delete*), execução (*execute*) e criação (*create*). Estas operações podem ser ordenadas. Dessa forma, se um usuário tem uma permissão de prioridade mais alta, isto implica que ele tem todas as permissões abaixo desta. Um usuário pode ter a ele associado um conjunto de permissões, o que pode ser chamado de autorizações positivas. Por outro lado, também pode ter um conjunto de operações negativas que são explicitamente proibidas de serem executadas.

Estes tipos de permissão são suficientemente abrangentes para serem usados em qualquer tipo de banco de dados. Assim sendo, serão igualmente adotados como base para as operações de acesso a bancos de dados geográficos.

3.3 Conclusão

A dissertação considera que a definição de autorização é generalizada pela frase:

> [Defina < *tipo de autorização* >
> sobre < *partição de dados no banco de dados* >]

Usando o modelo básico de autorização do capítulo 2 *tipo de autorização* é definido como sendo composto pelo par $< s, m >$, enquanto o é a *partição de dados no banco de dados*. Este capítulo apresenta os componentes $< s, m >$ que serão considerados na dissertação, e que são idênticos aos já propostos para bancos de dados convencionais. O próximo capítulo caracterizará a *partição de dados no banco de dados*, entendida como o resultado de uma consulta que retorna como resultado o objeto o (a partição).

Capítulo 4

Caracterização de consultas espaciais visando o controle de acesso

Este capítulo caracteriza as consultas espaciais visando o controle de acesso, classificando-as sob diferentes aspectos. Consultas espaciais retornam conjuntos de dados de um banco de dados espacial - o componente o da tupla $< s, o, m >$. Neste estudo foram considerados diversos tipos de consultas, tentando generalizar os requisitos, às vezes até com situações fictícias, sem se prender a nenhum tipo de aplicação em especial.

Rememorando o capítulo 2, os dados em um banco de dados geográfico podem ser caracterizados como objetos que possuem dois tipos de componentes:

- espaciais, que descrevem as características espaciais do objeto, em especial, localização e geometria. A geometria pode ser muitas vezes inferida a partir das coordenadas que fornecem a localização;
- descritivos, ou não espaciais, que descrevem as demais propriedades do objeto.

Para estudar acesso a bancos de dados geográficos, convém dividir o problema em:

- permissão para dados convencionais e
- permissão para dados espaciais.

A diferença entre as permissões para dados convencionais e dados espaciais está justamente no tipo de dado a ser acessado. A permissão para dados espaciais permite acessar diferentes tipos de objetos espaciais, tais como ponto, linha ou polígono. Uma permissão espacial está diretamente relacionada à consulta espacial que deve satisfazer. Por exemplo, a permissão "Ana tem acesso de leitura a todos os rios do estado de São Paulo", nada mais é do que uma permissão a todos os objetos resultantes da consulta "selecione todos

os rios do estado de São Paulo". Desta forma, a intenção deste capítulo é analisar as permissões através da caracterização das consultas espaciais que uma permissão pode envolver. Parte deste estudo baseou-se na sistematização de tipos de consultas definida por Sandro Matias em [Mat00] para o sistema BIOTA, acrescentando informações específicas para controle de acesso.

Este capítulo está organizado da seguinte forma. A seção 4.1 analisa os tipos de objetos envolvidos em uma consulta em um banco de dados geográfico para fins de controle de acesso. A seção 4.2 apresenta os operadores que se aplicam a estes objetos. A seção 4.3 exemplifica os relacionamentos existentes entre consultas e permissões. A seção 4.4 aborda consultas e permissões complexas e finalmente, a seção 4.5 faz uma breve conclusão deste capítulo.

4.1 Objetos sujeitos a controle de acesso

Esta seção caracteriza os diferentes tipos de objetos a serem utilizados nas permissões de acesso, a fim de facilitar a classificação das permissões que os utilizarão. Também é necessário definir as relações entre esses diferentes tipos de objetos.

Em um banco de dados não espacial, o controle de acesso é definido a partir de predicados sobre atributos descritivos. Como um banco de dados geográfico estende um banco de dados não espacial adicionando a parte espacial, o controle de acesso em bancos de dados geográficos pode ser definido também usando apenas características não espaciais. Assim sendo, a definição de controle de acesso pode ser:

- considerando apenas atributos descritivos;
- considerando apenas atributos espaciais; ou
- considerando ambos.

O primeiro tipo de caracterização já foi discutido na seção 2.1 na bibliografia correlata. A seguir será feita uma descrição dos dois outros tipos.

Os atributos espaciais a serem considerados nesta dissertação para efeito nas permissões de acesso são de três tipos:

- Ponto: objetos de geometria pontual são utilizados para definir diversos tipos de elementos do mundo real, entre eles: estabelecimentos comerciais (postos, supermercados, lojas), postes de uma rede externa de telefonia, árvores. Dependendo da escala utilizada, os pontos podem representar outros fenômenos. Por exemplo, em uma escala 1:1.000.000, cidades, pequenos bosques e vários tipos de superfície são representados por pontos.

- Linha: as linhas são geralmente utilizadas para representar objetos do mundo real que possam ser descritos por características lineares como ruas e trechos de logradouros de uma cidade, rodovias ou rios.

- Polígono: os polígonos representam áreas ou regiões como por exemplo: países, estados, cidades, bairros, áreas de enchente, área de preservação ambiental.

Os objetos espaciais podem ser considerados individualmente (**o** poste, **o** rio, **a** floresta) ou em conjuntos. Muitas vezes um SIG permite agrupar conjuntos de objetos segundo o fenômeno representado. Este agrupamento recebe o nome de camada (*layer*). Assim, freqüentemente se encontram na literatura referências a camadas de vegetação, hidrografia, etc. Cada camada corresponde à execução de uma consulta que retorna todos os objetos de um determinado tipo em uma região. Por exemplo, a camada hidrografia corresponde à consulta "retornar todos os objetos que descrevem cursos d'água em uma região".

Por razões de projeto de um banco de dados, normalmente objetos de um tipo são armazenados juntos (em relações ou em classes). Por exemplo, em um banco de dados relacional, objetos descrevendo rios estarão armazenados em uma relação Rios, objetos descrevendo postes estarão na relação Postes e assim por diante. Desta forma, a implementação de uma camada é freqüentemente mapeada na seleção de tuplas da relação correspondente. Isto causa problemas conceituais na especificação de permissões, pois há confusão de conceitos de SIG (camada) e bancos de dados (relação e atributo). Este texto considera o conceito de camada entendendo que ele tem subjacente uma consulta a conjuntos de objetos da mesma natureza ou relacionados semanticamente de alguma forma.

É importante salientar que existe uma hierarquia entre os tipos de objetos espaciais com relação à permissão, onde o nível básico é o ponto e o nível superior é o polígono. A figura 4.1 exibe a hierarquia dos objetos espaciais a partir do seu nível superior para o inferior: polígono $\rightarrow$ linha $\rightarrow$ ponto. Isto significa que ao se obter acesso a um polígono, obtém-se também o acesso a todos os objetos que estiverem contidos neste polígono. Quando se tem acesso a uma linha, tem-se acesso a todos os seus pontos. Entretanto, quando se tem acesso a um ponto, tem-se acesso somente àquele ponto, já que este é o nível básico. Pode haver problemas na determinação da permissão quando as relações espaciais entre objetos são mais complexas - por exemplo, quando um objeto não contém o outro totalmente. Estes problemas são discutidos no capítulo 5.

A figura 4.1 (parte direita) mostra um exemplo real de uma hierarquia entre objetos espaciais. O polígono representa o bairro de Barão Geraldo e dentro pode se ver a Avenida Um e o Posto BR. Neste caso, a especificação de uma permissão seria: "Ana tem acesso ao bairro de Barão Geraldo". Isto significa que Ana tem acesso a todos os objetos do tipo

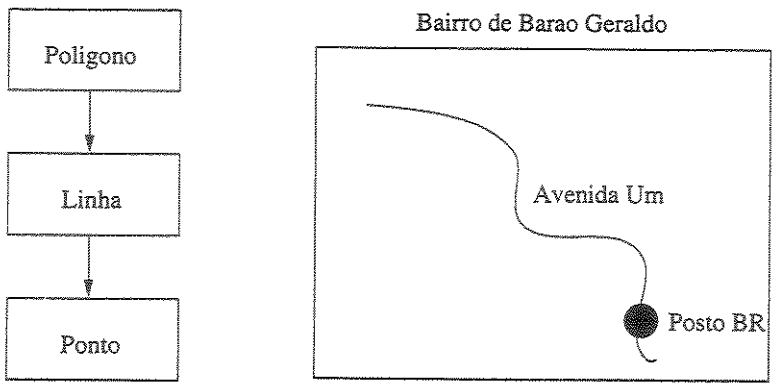


Figura 4.1: Hierarquia dos objetos espaciais e um exemplo.

ponto e linha que fazem parte deste polígono. Assim, Ana pode acessar tanto a Avenida Um quanto o posto BR. No exemplo relativo a Ana, o polígono representa os limites de um objeto do mundo real (o bairro de Barão Geraldo). No entanto, polígonos podem ser especificados por consultas de usuário e não corresponderem exatamente a nenhum objeto real. Este caso, bastante comum em situações de controle de acesso, corresponde a situações em que algum usuário autorizado define uma região dinamicamente sobre um mapa - por exemplo, traçando os limites de um retângulo sobre uma região. Neste caso, o retângulo é o polígono e todos os objetos nele inscritos estarão sujeitos à regra de acesso. O problema ocorre quando este polígono arbitrário corta objetos ao meio - por exemplo, o retângulo definido cobre parcialmente dois bairros vizinhos, ou mesmo apenas parte de uma quadra. Este tipo de problema é estudado na seção 5.2.

4.2 Operadores espaciais

Além de predicados convencionais, as consultas em bancos de dados geográficos utilizam operadores espaciais. Estes relacionamentos espaciais são classificados em topológicos, métricos e direcionais [Gut94].

Os relacionamentos topológicos, como "dentro de" e "superposto a", retornam um valor booleano e são invariantes face a transformações biunívocas e bicontínuas, como transformações de escala, translação e rotação. A definição de um conjunto mínimo de relacionamentos topológicos é objeto de muito debate na literatura. Freeman [Fre75] define até treze tipos de relacionamentos: "à esquerda de", "à direita de", "acima", "abaixo", "atrás", "próximo a", "longe de", "ao lado de", "tocando", "dentro de", "fora de" e "entre". Há ainda os que listam seis [Feu93] e outros que listam oito [EH90]. Muitas

dessas definições são conflitantes quando tentam definir de que tipo são os relacionamentos. Aqui serão considerados apenas os cinco relacionamentos definidos por Clementini e outros [ECvO93], que mostram que estes relacionamentos são suficientes para representar predicados topológicos binários. São eles: dentro de, superposto a, tocando, cruzando e disjunto (*in*, *overlap*, *touch*, *cross* e *disjoint*). Para aplicações de rede, deve-se ainda considerar a direção de fluxo, o que é previsto por dois operadores unários adicionais "de" e "para" (*to*, *from*).

Exemplos de consultas que definem acesso usando relacionamentos topológicos são:

- Maria tem acesso a todos os supermercados da cidade de Campinas - todos os pontos dentro de uma área - operador "dentro de"
- Maria tem acesso às áreas alagadas do bairro Cambuí - todas as áreas superpostas a outra área - operador "superposto a"
- Maria tem acesso a todas as ruas contíguas ao bairro Cambuí - todas as linhas que tocam uma área - operador "tocando"
- Maria tem acesso a todas as ruas que cruzam com a Av. Brasil - todos as linhas que cruzam uma outra linha - operador "cruzando"
- Maria tem acesso a todos os bairros que não fazem limite com o bairro Santa Genebra - todas as áreas disjuntas de uma outra área - operador "disjunto"

As operações métricas são as operações que envolvem medidas como distância ou perímetro. Como exemplo, podemos citar:

- Maria tem acesso a todas as cidades que distam no máximo 50 km de Campinas - todas as áreas que ficam a uma determinada distância de outra área

Os relacionamentos direcionais, como "acima de" e "perto de", retornam um valor booleano e descrevem como os objetos espaciais estão posicionados uns em relação aos outros. A definição de um relacionamento direcional em geral envolve um marco de referência, um objeto de referência e o objeto em questão. O marco de referência determina a direção na qual o objeto em questão está localizado em relação ao objeto de referência. Estudos sobre sentenças espaciais em linguagem natural revelam que os relacionamentos direcionais dependem de aspectos cognitivos, que variam culturalmente. Assim, estes relacionamentos são considerados *fuzzy* e, ao contrário dos topológicos, não são invariantes. Dentre os operadores espaciais direcionais, podemos destacar: "acima de", "abaixo de", "à esquerda", "à direita", "à leste", "à oeste", "ao sul", "ao norte". Alguns exemplos são:

- Maria tem acesso a todas as ruas acima da estação de trem - operador *fuzzy* "acima de"
- Maria tem acesso às cidades ao sul de Campinas - operador *fuzzy* "ao sul de"
- Maria tem acesso aos rios perto da cidade de São Paulo - operador *fuzzy* "perto de"
- Maria tem acesso a todos os bairros ao norte do bairro Cambuí - operador *fuzzy* "ao norte de"

Estes exemplos indicam a dificuldade em se determinar o resultado das consultas que envolvem operadores *fuzzy*, já que estes dependem de aspectos cognitivos que podem variar culturalmente. Existem ainda outros operadores *fuzzy* como: "longe de", "ao redor", "ao lado de" e "atrás". Os operadores direcionais não serão considerados nesta dissertação, tendo em vista a complexidade em se determinar a sua semântica.

4.3 Construindo consultas - objetos e predicados

Esta seção combina os conceitos das duas seções anteriores organizando as consultas para efeito de estudo de controle de acesso.

A tabela 4.1 mostra como consultas espaciais envolvem relacionamentos entre diferentes tipos de objetos espaciais. Estas consultas retornam um resultado, que é o alvo do controle de acesso, o objeto o da tupla $< s, o, m >$.

	Ponto	**Linha**	**Polígono**
Ponto	C_1	C_2	C_3
Linha	C_2	C_4	C_5
Polígono	C_3	C_5	C_6

Tabela 4.1: Objetos espaciais e seus interrelacionamentos.

A apresentação do caracter C em uma célula da tabela representa um tipo de consulta. A cada resultado de consulta podem ser associados diferentes tipos de permissões. Por exemplo, C_2 corresponde a consultas envolvendo relacionamentos entre pontos e linhas ou linha e pontos.

Consultas podem retornar atributos descritivos, espaciais ou ambos. Podem também retornar objetos espaciais e não espaciais.

A consulta C_x "Quais os assinantes cadastrados no banco de dados" retorna objetos não espaciais (assinantes).

A consulta C_y "Quais os tipos dos cabos instalados no bairro Cambuí" retorna atributos descritivos (tipo de cabo).

A consulta C_z "Supermercados com mais de 5 telefones instalados" retorna objetos espaciais (supermercados), supondo que têm um componente espacial.

As consultas C_x e C_z utilizam predicados não espaciais, enquanto a consulta C_y utiliza predicado espacial. O estudo de objetos do banco de dados visando acesso precisa assim considerar consultas quando a:

- resultado - objeto espacial ou não espacial, ou parte de um objeto;
- predicado - espacial, não espacial ou ambos.

Seja por exemplo a consulta C_z "Supermercados com mais de 5 telefones instalados". Um exemplo de permissão envolvendo C_z seria "Ana pode modificar dados relativos aos supermercados com mais de 5 telefones instalados", sendo:

s:	Ana
o:	pontos (supermercados)
m:	escrever
predicado:	sobre atributos descritivos (número de telefones de um supermercado)

Outro exemplo seria "Todos os supervisores podem ler dados da Rua Marquês de Olinda", onde:

s:	todos os supervisores
o:	linha (a rua)
m:	leitura
predicado:	descritivo (nome de rua = Marquês de Olinda)

Este tipo de raciocínio, separando a definição da permissão da definição do objeto pode ser repetido para todos os demais exemplos que se seguem, sendo, por esta razão omitida de agora em diante. Os exemplos cobrem todas as células da tabela e se concentram em predicados espaciais. Predicados não espaciais não representam novidade do ponto de vista de consultas, sendo por este motivo omitidos.

Exemplo 1: Ponto X Ponto

C_1: selecione todos os supermercados a menos de 1000 m do Supermercado Barão
o: conjuntos de pontos (supermercados)
predicado métrico: distância (o, ponto específico)

Exemplo 2: Linha X Ponto ou Ponto X Linha

C_2: selecione as ruas que contêm um posto BR
o: linhas (ruas)
predicado topológico: "dentro de" (o, ponto)

C_2: selecione todos os postos a menos de 3 km da Rua José Paulino
o: pontos (postos)
predicado métrico: distância (o, linha específica)

Exemplo 3: - Polígono X Ponto ou Polígono X Ponto

No exemplo a seguir, o predicado tanto pode ser interpretado como um predicado envolvendo um atributo não espacial ou um predicado que utiliza o operador topológico "dentro de". Tudo depende do esquema do banco de dados. No primeiro caso, é necessário que os postos tenham um atributo que represente o nome da cidade onde o posto se localiza. Já no caso 2, é preciso que haja um polígono que represente a cidade de Campinas e que os pontos de postos estejam dentro deste polígono.

C_3: selecione todos os postos da cidade de Campinas
o: pontos (postos)
predicado topológico: "dentro de" (o, polígono), ou predicado não espacial

C_3: selecione uma área de 5 km ao redor da Central Telefônica do Castelo
o: polígono (área)
predicado métrico: distância (perímetro o, ponto específico)

Exemplo 4: Linha X Linha

C_4: selecione as ruas que cruzam com a Rua José Paulino
o: linhas (ruas)
predicado topológico: cruza (o, linha específica)

Exemplo 5: Linha X Polígono ou Polígono X Linha

Da mesma maneira que o primeiro exemplo C_3, o exemplo a seguir pode ser considerado como um predicado baseado em atributo não espacial ou predicado topológico "dentro de".

C_5: selecione as ruas da cidade de Campinas

o: linhas (ruas)
predicado topológico: contido (o, área) ou predicado não espacial

C_5: selecione os municípios atravessados pelo Rio Tietê
o: polígonos (municípios)
predicado topológico: cruza (o, linha específica)

Exemplo 6: Polígono X Polígono

C_6: selecione os bairros que fazem parte da Zona Norte
o: polígonos (bairros)
predicado topológico: contido (o, polígono específico)

4.4 Consultas e permissões complexas

Para cada tipo de consulta pode haver vários tipos de permissões. O sujeito da permissão pode ser especificado diretamente, como nos exemplos do início da seção anterior: "Ana pode modificar dados relativos aos supermercados com mais de 5 telefones instalados".

Uma outra hipótese é que a permissão seja especificada a partir de outra permissão, por exemplo: "Todas as pessoas que têm acesso de leitura aos supermercados de Campinas também podem modificar dados relativos aos supermercados com mais de 5 telefones instalados". Neste caso, a regra de acesso pode ser decomposta recursivamente da seguinte forma:

s: todas as pessoas que têm acesso de leitura aos supermercados de Campinas
o: pontos (supermercados)
m: escrita

s, por sua vez, é também uma regra de acesso, decomposta por:

s_1: todas as pessoas
o_1: pontos (supermercados) resultantes da consulta "supermercados de Campinas"
m_1: leitura

Em outras palavras, a regra final é $<< s_1, o_1, m_1 >, o, m >$, sendo s especificado indiretamente. Da mesma forma, objetos podem ser definidos indiretamente: "Ana tem acesso de escrita a todos os objetos acessáveis por supervisores".

s: Ana
o: todos os objetos acessáveis por supervisores
m: escrita

Supondo-se que haja outra permissão: "Os supervisores têm acesso de escrita ao supermercados com mais de 5 telefones instalados".

s_2: supervisores
o_2: supermercados com mais de 5 telefones instalados
m_2: escrita

Esta regra corresponde portanto a $< s, < s_2, o_2, m_2 >, m >$ Uma regra que envolva as duas regras seria: "Todas as pessoas que têm acesso de leitura aos supermercados de Campinas têm igualmente acesso de escrita a todos os objetos acessáveis por supervisores". Esta regra é processada como $<< s_1, o_1, m_1 >, < s_2, o_2, m_2 >, m >$.

A composição de permissões, desta forma, pode ser resolvida dentro do modelo proposto através da execução apropriada de consultas e identificação de objetos. Por este motivo, não se considera aqui que a composição de permissões ofereça problemas adicionais do ponto de vista de identificação da tupla $< s, o, m >$. Sendo assim, para o restante deste trabalho, apenas elementos simples serão considerados, facilitando as explicações do texto.

4.5 Conclusão

Este capítulo caracterizou as consultas visando o controle de acesso e mostrou a distinção entre as consultas e permissões usando os operadores espaciais possíveis para estas consultas. Também sistematizou os interrelacionamentos existentes entre os objetos espaciais utilizados em consultas e, conseqüentemente, em permissões de controle de acesso. O próximo capítulo combina a sistematização $< s, o, m >$ feita nos capítulos 3 $< s, m >$ e 4 $< o >$ para propor o modelo de acesso.

Capítulo 5

Modelo de autorização para bancos de dados geográficos

Os capítulos 3 e 4 discutem detalhes da tripla $< s, o, m >$ para definir autorizações. Esta tripla foi inicialmente proposta no modelo de controle de acesso seletivo de Griffths e Wade [GW76] e é a base da proposta da dissertação.

Este capítulo define os principais itens do modelo de autorização aqui proposto levando-se em consideração as características dos dados espaciais, a saber:

- a determinação da granularidade de acesso;
- estruturas para representar a autorização (semântica formal de representação);
- um conjunto de políticas para gerenciar e administrar autorizações;
- algoritmos para analisar requisições de acesso baseando-se nas autorizações.

O capítulo está organizado da seguinte forma. A seção 5.1 apresenta uma visão geral do problema de acesso, dividindo-o em direito a acesso (gerenciamento de autorizações) e permissão de acesso (gerenciamento de concorrência). O modelo trata apenas do primeiro problema, assim sendo a seção 5.2 detalha questões de conflito no direito a acesso. A seção 5.3 apresenta o modelo de autorização dividindo-o em seus principais componentes. As seções 5.3.1, 5.3.2, 5.3.3 e 5.3.4 especificam a granularidade, os sujeitos, tipos de acesso, objetos, regras de autorização e políticas adotadas. A seção 5.4 discute o problema de armazenamento das autorizações. A seção 5.5 propõe uma arquitetura básica para o controle de acesso no que diz respeito a direito de acesso. A seção 5.6 define os algoritmos para verificar um pedido de acesso baseando-se nas regras de autorização existentes no banco de dados e para atualização da base de regras de autorização. A seção 5.7 apresenta uma arquitetura opcional tendo o mecanismo de controle de acesso como uma camada

SQL a fim de evitar acessos diretos aos dados. Finalmente, a seção 5.8 apresenta as conclusões do capítulo.

5.1 Problema geral do controle de acesso

A dissertação propõe que o controle de acesso em bancos de dados geográficos seja entendido com um processo em duas etapas principais, a partir de uma solicitação do usuário em um dado momento:

1. Direito a acesso: verificar se o usuário tem direito a acesso, a partir de regras de autorização previamente definidas;

2. Permissão de acesso: caso positivo, verificar se o acesso pode ser concedido, em função de eventuais problemas de concorrência, naquele momento.

A dissertação se atém à primeira etapa, pressupondo que questões de concorrência são tratadas pelo SGBD. Esta primeira etapa é esboçada na seção 5.1.1 e detalhada na seção 5.2. Alguns dos problemas da etapa (2) são mencionados na seção 5.1.2. A seção 5.5 propõe uma arquitetura básica para o controle de acesso visando somente a etapa (1).

5.1.1 Gerenciamento de direito a acesso para dados geográficos

Os problemas de direito a acesso no caso de dados geográficos estão resumidos graficamente na figura 5.1. A figura mostra que um sujeito s pediu acesso a um conjunto de objetos delimitados pela área PA (pedido de acesso). Este conjunto pode conter, inclusive, um único objeto pontual. O banco de dados, por sua vez, já tem armazenado diferentes regras de autorização $< s, o, m >$ (R_1, ..., R_4). O problema de direito a acesso consiste em determinar se o pedido de acesso PA é aceitável ou se é conflitante com as regras existentes.

O conflito também pode ocorrer entre regras. Suponha que PA na figura 5.1 seja uma regra R_5 a ser inserida. Aqui também é necessário determinar se a nova regra R_5 é conflitante com as outras já existentes. Neste caso, os conflitos devem ser resolvidos no momento da inserção desta nova regra. No primeiro caso, os conflitos devem ser resolvidos no momento da validação de um pedido de acesso.

Um primeiro problema, que não será tratado na dissertação, é quando um conjunto de regras é inconsistente (por exemplo, se alguma regra no conjunto for a negação de outra regra do mesmo conjunto). Um outro problema é determinar quando o conjunto de regras de controle de acesso apresenta conflitos com um pedido de acesso. Para analisar esta situação, a dissertação considerará apenas autorizações positivas.

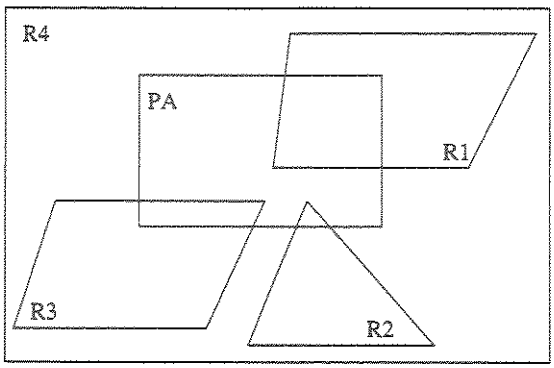


Figura 5.1: Pedido de acesso.

Definição 1: Sejam duas regras representadas pela tuplas $R_1 =< s_1, o_1, m_1 >$ e $R_2 =< s_2, o_2, m_2 >$. Dizemos que R_1 e R_2 podem entrar em conflito de direito de acesso se $s_1 \neq s_2$ e $o_1 \bigcap o_2 \neq 0$ (ou seja, sujeitos diferentes acessando potencialmente um mesmo conjunto de objetos). Lembrando que R_1 pode também ser entendida como um pedido de acesso.

No caso de objetos ou componentes não espaciais, objetos podem ter granularidades variadas (atributo, tupla, relação). Mesmo no caso não espacial, pode haver problemas de conflito, por exemplo, se o_1 é uma relação r_1 e o_2 é um conjunto de tuplas de r_1. O problema não é discutido na literatura, pois em geral a hipótese subjacente é que o próprio mecanismo de trancas do controle de concorrência resolverá a questão.

No caso de bancos de dados geográficos, no entanto, o_1 e o_2 podem também representar conjuntos de dados espaciais em que a verificação de conflitos exige realizar consultas complexas. Por exemplo, o_1 pode ser um polígono e o_2 uma linha parcialmente contida no polígono. É preciso, neste caso, decidir se s_1 tem permissão à linha em sua integralidade (ou seja, ultrapassando os limites do polígono) ou apenas à parte da linha (exigindo corte de objetos) ou até mesmo proibir o acesso. Este tipo de questão não ocorre para componentes descritivos.

Um outro tipo de conflito é quando há um pedido de acesso $PA =< s_1, o_1, m_1 >$ e uma regra $R_2 =< s_2, o_2, m_2 >$ para o mesmo sujeito com o mesmo tipo de operação. Neste caso, é preciso verificar o relacionamento entre os objetos o_1 e o_2.

Definição 2: Seja um pedido de acesso representado pela tupla $PA =< s_1, o_1, m_1 >$ e uma regra de autorização existente no banco de dados $R_2 =< s_2, o_2, m_2 >$. Dizemos que PA e R_2 podem entrar em conflito de direito de acesso se $s_1 = s_2$, $m_1 = m_2$ e $o_1 \neq o_2$, mas $o_1 \bigcap o_2 \neq 0$ (ou seja, existe uma autorização para s_1 com o mesmo tipo de operação m_1, mas objetos distintos que fazem intersecção).

Suponha um exemplo onde s_1 deseja acessar o_1, que é uma linha totalmente contida no polígono o_2, a qual s_1 tem acesso. Neste caso, a política mais factível é conceder o direito de acesso, já que a linha está totalmente contida no polígono. Entretanto, há conflitos especialmente quando o_1 estiver parcialmente contido em o_2.

A seção 5.2 tratará destes problemas específicos e da solução proposta para cada caso de intersecção de componentes espaciais.

5.1.2 Gerenciamento de concessão de acesso

Após a validação do direito a acesso (etapa 1), deve-se ter uma maneira de conceder o acesso garantindo a consistência dos dados. Se o mecanismo permitir mais de um acesso simultâneo, os dados podem ficar inconsistentes. Entretanto, há aplicações em que é estritamente necessário permitir acessos simultâneos. Assim sendo, existem duas alternativas: evitar acessos simultâneos ou permitir os acessos sem causar inconsistências no banco de dados. Um exemplo real é que duas equipes de projeto de expansão de rede externa podem estar tentando acessar a mesma área.

A partir do momento em que o direito a acesso foi garantido, a concessão do acesso passa a ser do escopo de algum outro mecanismo. Uma solução que permite acessos concorrentes é utilizar um mecanismo de versões. Dias e outros [EDM95] propuseram um mecanismo de versões para garantia de consistência para bancos de dados geográficos, onde há vários usuários tentando acessar a mesma base de dados ao mesmo tempo.

O modelo de autorização aqui proposto não tratará deste problema. Ele pode ser acoplado ao modelo de Dias ou outro semelhante em uma futura extensão.

A solução de Dias e outros [EDM95] propõe o uso de um mecanismo de versões para ambientes mistos de projeto e operação. Nestes ambientes existe a presença de transações longas e aninhadas, bem como transações curtas. Este trabalho de Dias apresentou uma implementação do conceito de versões na garantia de consistência de atualizações de objetos complexos quando o banco de dados está sendo usado tanto por projetistas operando com transações longas e aninhadas como por operadores transacionando em regime *on-line* com transações curtas. Isto foi implementado no sistema SAGRE. A solução, além de garantir a viabilidade de uma implementação prática, permite a convivência controlada de atualizações de curta duração e de longa duração envolvendo mudanças condicionais,

isto é, mudanças que podem ou não ser efetivadas.

No caso do SAGRE, vários usuários trabalham cooperativamente para produzir um projeto. Neste ambiente, cada projetista cria uma parte do todo em várias etapas inclusive analisando alternativas. Existe ainda o agravante de que projetos de redes de telecomunicações geralmente atuam sobre uma área geográfica onde já existe alguma rede já em operação. Assim, os projetistas trabalham sobre uma situação existente que não pode ser congelada uma vez que as redes implantadas são operadas de forma *on-line* - os dados devem estar disponíveis com agilidade e enquanto se projeta uma rede numa determinada região, ela pode estar sendo atualizada. Estas alterações podem ser provocadas por acidentes da natureza (um raio derrubando um poste) ou pressão da demanda por serviços (um pequeno aumento enquanto se espera pela expansão projetada).

5.2 Resolução de conflitos entre componentes espaciais

Esta seção discute problemas de direito de acesso quando ocorrem os dois tipos de conflitos (Definição 1 e Definição 2) definidos na seção 5.1.1.

A seção 5.2.1 apresenta as soluções possíveis para os conflitos segundo a Definição 1: duas regras $R1 =< s1, o1, m1 >$ e $R2 =< s2, o2, m2 >$ apresentam conflitos relativos a intersecções espaciais tendo sujeitos distintos.

A seção 5.2.2 apresenta as soluções possíveis para os conflitos segundo a Definição 2: um pedido de acesso $PA =< s1, o1, m1 >$ e uma regra $R2 =< s1, o2, m1 >$ apresentam conflitos relativos a intersecções espaciais tendo o mesmo sujeito e mesmo modo de acesso.

5.2.1 Conflitos do tipo 1

5.2.1.1 Intersecção de polígonos

Considere a área de intersecção $(o1 \bigcap o2)$ entre os polígonos o_1 e o_2, que corresponde à área mais escura. Um exemplo de uma situação real para este caso é quando duas equipes de projetos de expansão de rede externa de telefonia têm direito a acessar regiões não disjuntas de uma cidade.

O caso geral de intersecção de polígonos portanto ocorre quando há duas regras $R_1 =< s_1, o_1, m >$ e $R_2 =< s_2, o_2, m >$ e se deseja saber se s_1 pode ou não acessar a área $(o_1 \bigcap o_2)$? As seguintes opções podem ser consideradas:

1. sim, s_1 pode acessar qualquer objeto de o_1, mesmo que este também pertença a algum outro polígono;

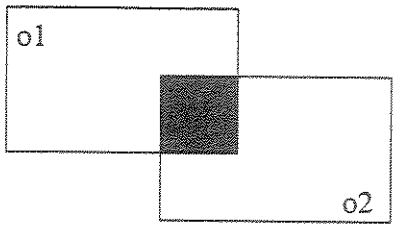


Figura 5.2: Intersecção de polígonos.

2. sim, somente se existir também uma autorização $< s_1, o_2, m >$, que autorize s_1 a acessar dados também contidos no polígono o_2;

3. sim, s_1 pode acessar o_1, contanto que não exista uma autorização negativa do tipo $< s_1, o_2, m, - >$, que proíba s_1 de acessar dados do polígono o_2, considerando-se o caso em que existam autorizações negativas;

4. não, a situação não ocorre porque no banco de dados não se aceitam regras onde haja intersecção de polígonos;

5. não, não é permitido o acesso a áreas que também pertençam a outros polígonos.

A opção (4) elimina o problema, já que esta situação de intersecção não ocorreria. Entretanto, esta não é a mais indicada, já que em várias situações do mundo real existem áreas que se sobrepõem, por exemplo, áreas de cobertura de telefonia. Uma determinada região pode estar coberta por dois setores de estações rádio base diferentes.

A opção (1) pode causar falhas e inconsistências, já que permite acessar dados que também pertencem à área o_2 e existe a possibilidade de que não se deseje que s_1 acesse dados de o_2. A opção (3) resolve este problema, obrigando que se defina explicitamente que s_1 não deve acessar dados de o_2 através de uma autorização negativa. Entretanto, nem todas os sistemas aceitam autorizações negativas e o gerenciamento de autorizações negativas é muito mais caro.

A opção (2) também possui uma desvantagem, pois para poder acessar $(o_1 \bigcap o_2)$ o sujeito precisa ter autorização tanto para a área o_1 quanto para a área o_2. Existe também nesse caso a possibilidade de se desejar que s_1 acesse somente os dados de o_1, mas não os

de o_2, assim nunca se poderia ter a situação: s_1 pode acessar dados de o_1, mas não os de o_2 e ainda poder acessar dados da área $(o_1 \bigcap o_2)$.

A opção (5) diz que s_1 pode acessar todos os dados de o_1 contanto que estes não estejam contidos em outros polígonos. O problema aqui é que os usuários nunca podem acessar áreas que se intersectam, impossibilitando todas as operações sobre objetos que pertençam a áreas comuns entre dois ou mais polígonos.

Não existe, desta forma, solução ideal. A dissertação propõe a adoção da solução (1), por ser a mais fácil de implementar, mas a política de acesso depende de cada caso.

Os casos de intersecção de polígonos podem ainda ser estendidos para outras situações, como inclusão de um polígono em outro e múltiplas intersecções - vide por exemplo as figuras 5.7 e 5.3. As opções para continência de polígonos podem ser entendidas como sendo as mesmas para intersecção.

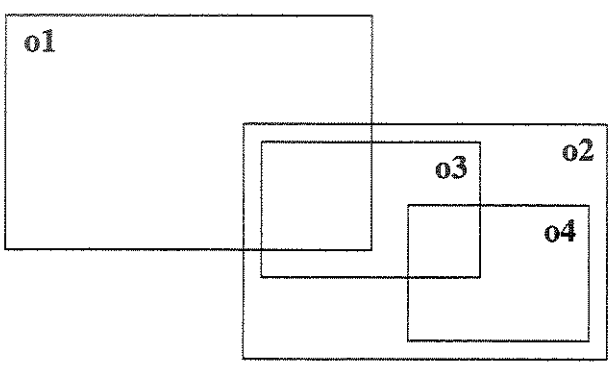


Figura 5.3: Intersecção e continência de polígono.

Generalizando, seja a situação da figura 5.3 para intersecção e continência em que o sujeito s_1 quer acessar dados da região o_1. O que se pode dizer do acesso de s_1 em relação às áreas $(o_1 \bigcap o_2)$ e $(o_1 \bigcap o_3)$. Recaímos nos mesmos casos mencionados anteriormente para intersecção.

5.2.1.2 Intersecção de linhas e polígonos

O segundo tipo de intersecção espacial envolve linhas e polígonos. Seja a situação onde um uma linha L está parcialmente contida em um polígono o_1 e o sujeito s_1 tem permissão de acesso ao polígono o_1 através da regra $R_1 =< s_1, o_1, m >$) em conflito com uma outra regra $R_2 =< s_2, L, m >$. O que se pode dizer do acesso de s_1 a L? Como no caso anterior, existem várias opções.

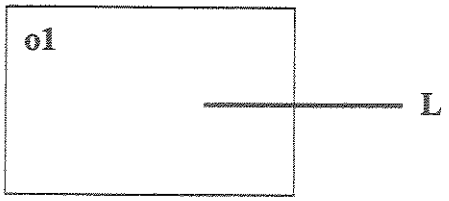


Figura 5.4: Superposição parcial de linha e polígono.

1. sim, s_1 pode acessar a parte de L que está contida em o_1. Entretanto, isso exige que L seja dividido em duas partes: L_1, a parte interna a o_1 e L_x, a parte externa a o_1 (figura 5.5). O problema pode ser tratado do ponto de vista da situação anterior envolvendo polígonos - basta supor a existência de um polígono adicional X que contenha L_x e que não tenha intersecção com o_1;

2. s_1 pode acessar L desde que exista uma autorização com outra granularidade (linha, neste caso) que permita s_1 acessar o objeto $L < s_1, L, m >$;

3. s_1 pode acessar L em toda a sua extensão desde que exista também uma autorização $< s_1, X, m >$, que autorize s_1 a acessar dados do polígono X, externo a o_1 e que englobe L_x. Neste caso, existe o problema de se querer acessar o_1, mas não X e ainda querer acessar o objeto L que atravessa os dois polígonos.;

4. s_1 pode acessar L, contanto que não exista uma autorização negativa do tipo $< s_1, X, m, - >$, que proíba s_1 de acessar dados do polígono X;

5. não se aceitam regras que envolvam objetos parcialmente contidos em polígonos;

6. não se pode acessar objetos parcialmente contidos em polígonos, apenas os objetos totalmente contidos.

A opções (5) e (6) são as mais fáceis de se implementar, pois essas duas políticas simplesmente não tratariam esses casos. Entretanto, isto não é o mais indicado, pois não estaria refletindo as necessidades do mundo real, já que várias aplicações reais exigem objetos parcialmente contidos em polígonos, por exemplo, ruas e avenidas que atravessam vários bairros distintos.

A opção (2) exige que se defina explicitamente uma autorização ao objeto parcialmente contido. Neste caso, é possível que não se deseje permitir o acesso ao objeto todo e sim a

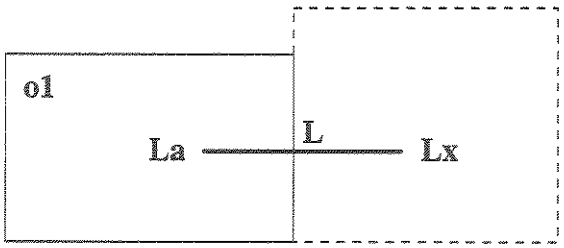


Figura 5.5: Divisão de linha.

apenas parte do objeto que está em o_1, além do fato do sistema ter que suportar permissões com várias granularidades.

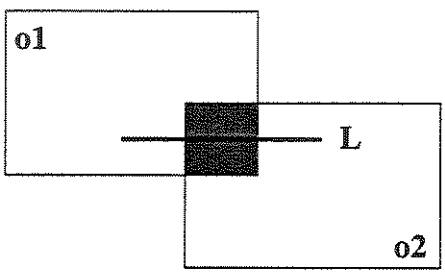


Figura 5.6: Superposição parcial de linha e intersecção de polígonos.

A opção (3) também não condiz com a realidade, pois existem casos em que não se deseje acessar os dados do polígono X, apenas os de o_1. A opção (4) resolve este problema usando a autorização negativa, mas isso exige que a aplicação aceite autorizações negativas, com os problemas já mencionados.

Assim, a abordagem mais factível seria a opção (1), que permite cortar objetos ao meio. Entretanto, isso aumenta a complexidade do sistema, exigindo que se tenha operações que cortem os objetos exatamente na divisão dos polígonos para o caso em que os polígonos se toquem. Em casos reais L_x existe e pode mesmo ter intersecção com o_1.

A situação mais geral combina intersecção de linhas e polígonos e está retratada na figura 5.6. Neste caso, a solução depende também da solução adotada para intersecção de polígonos.

5.2.2 Conflitos do tipo 2

O segundo tipo de conflito ocorre quando um sujeito s_1 pede acesso $PA =< s_1, o_1, m_1 >$ a um objeto o_1 e na base de regras existe apenas uma regra $R_2 =< s_1, o_2, m_1 >$ que autoriza s_1 a acessar o objeto o_2. Para validar o pedido de acesso PA, é preciso verificar o relacionamento espacial existente entre os objetos o_1 e o_2.

Generalizando, este tipo de conflito pode ser entendido como tendo apenas dois casos: objetos totalmente contidos em outro ou objetos parcialmente contidos em outro, ou seja, o_1 total ou parcialmente contido em o_2.

5.2.2.1 Objetos totalmente contidos

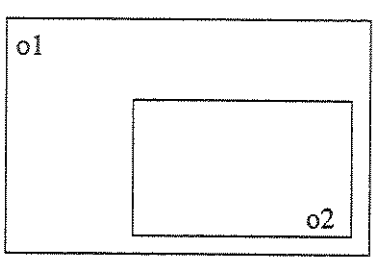


Figura 5.7: Continência de polígono em polígono.

A seção 4.1 define uma hierarquia para objetos espaciais com relação à permissão, onde o nível básico é o ponto e o nível superior é o polígono. A hierarquia a partir do seu nível superior para o inferior corresponde a: polígono → linha → ponto. Isto significa que ao se obter acesso a um polígono, obtém-se também o acesso a todos os objetos que estiverem totalmente contidos neste polígono. Quando se tem acesso a uma linha, tem-se acesso a todos os seus pontos.

Portanto, pode-se definir regras de inferência para hierarquias de objetos totalmente contidos um no outro. A existência da autorização $< s_1, o_2, m_1 >$ permite inferir a autorização $< s_1, o_1, m_1 >$, não sendo necessário defini-la explicitamente, se o_1 estiver

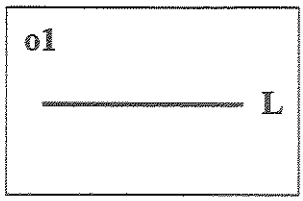


Figura 5.8: Continência de linha em polígono.

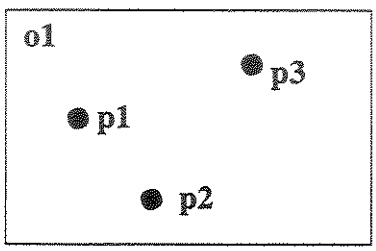


Figura 5.9: Continência de pontos em polígono.

totalmente contido em o_2. Caso não se deseje que s_1 acesse o_1, embora possa acessar o_2, deve-se definir explicitamente uma autorização negativa $< s_1, o_2, m_1, - >$. Para os casos de continência temos as seguintes situações e alguns exemplos representados nas figuras 5.7, 5.8 e 5.9.

- polígono contido em polígono;
- linha contida em polígono;
- linha contida em linha;
- ponto contido em polígono;
- ponto contido em linha;
- ponto contido em ponto.

5.2.2.2 Objetos parcialmente contidos

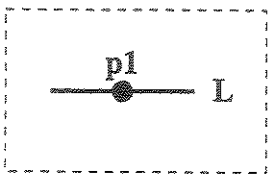


Figura 5.10: Ponto sobre linha.

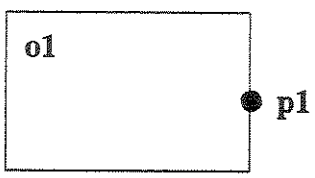


Figura 5.11: Intersecção de ponto e polígono.

A questão é o que fazer quando o objeto o_1 estiver parcialmente contido em o_2. Deve-se ou não permitir o acesso de s_1 a o_1? Neste caso, existem várias opções:

1. sim, s_1 pode acessar o objeto o_1, mesmo que este esteja parcialmente contido;

2. s_1 pode acessar parte do objeto o_1 que esteja contido em o_2, exigindo cortar objetos ao meio;

3. sim, somente se existir também uma autorização $< s_1, o_1, m_1 >$ na base de regras, que autorize s_1 a acessar o_1 explicitamente;

4. sim, somente se existir também uma outra autorização $< s_1, o_3, m_1 >$ na base de regras, onde o_3 contenha a outra metade de o_1;

5. sim, s_1 pode acessar o_1, contanto que não exista uma autorização negativa do tipo $< s_1, o_1, m_1, - >$, que proíba s_1 de acessar o_1, considerando-se o caso em que existam autorizações negativas;

6. não, a situação não ocorre porque no banco de dados não se aceitam regras onde haja objetos parcialmente contidos em outros;

7. não, não é permitido o acesso a objetos parcialmente contidos em outros.

Alguns exemplos de objetos parcialmente contidos em outros podem ser representados pelas figuras 5.4 e 5.2.

Quando pontos encontram-se na fronteira de polígonos (figura 5.11) ou sobre linhas (figura 5.10), embora estes estejam na verdade parcialmente contidos, pode-se considerar como um objeto totalmente contido, já que não faz muito sentido dividir um objeto tão pequeno. Além do fato de pontos serem o objeto de menor granularidade na hierarquia definida em 4.1.

5.3 Modelo de autorização para dados geográficos

As seções seguintes apresentam especificações dos principais componentes do modelo de autorização: a granularidade, sujeito, objeto, modo de acesso, regras de autorização e políticas adotadas.

5.3.1 Granularidade

Um dos itens mais importantes no modelo de controle de acesso é a questão da granularidade, pois dela depende todo o resto. A granularidade é definida pelo próprio objeto a ser acessado. Como vimos no capítulo 4, deve ser possível acessar objetos do tipo ponto, linha, polígono e conjuntos destes. Ao permitir várias granularidades, aumenta-se a complexidade do sistema. Quanto menor a granularidade, maior o número de permissões a serem definidas. Por exemplo, definir isoladamente as permissões para cada poste (supondo-se que um poste seja representado por um ponto) de uma cidade envolve muito mais detalhes de dados e controle do que definir uma única permissão para todos os objetos do tipo Poste. Como ocorre freqüentemente em problemas envolvendo sistemas de bancos de dados, é necessário estabelecer um compromisso entre espaço ocupado e desempenho. No caso, espaço ocupado refere-se ao armazenamento de informações de controle versus o desempenho do mecanismo de controle de acesso.

A granularidade mínima corresponde exatamente ao objeto a ser acessado. Aqui, trata-se de elementos espaciais. Como os objetos podem ser especificados diretamente, através de seus identificadores, ou indiretamente como resultantes de uma consulta, pode-se armazenar as autorizações usando consultas.

A solução proposta considera duas representações alternativas de regras: $< s, o, m >$ e $< s, C, m >$. A primeira representação referencia explicitamente o identificador do objeto (por exemplo, o ponto referente a um poste específico, a linha referente a uma rua, o polígono referente a um bairro). Neste caso, a autorização recai sobre um objeto individual do banco de dados. A segunda representação contém uma especificação da consulta em SQL espacial que define os objetos a serem controlados pela regra (vide capítulo 4).

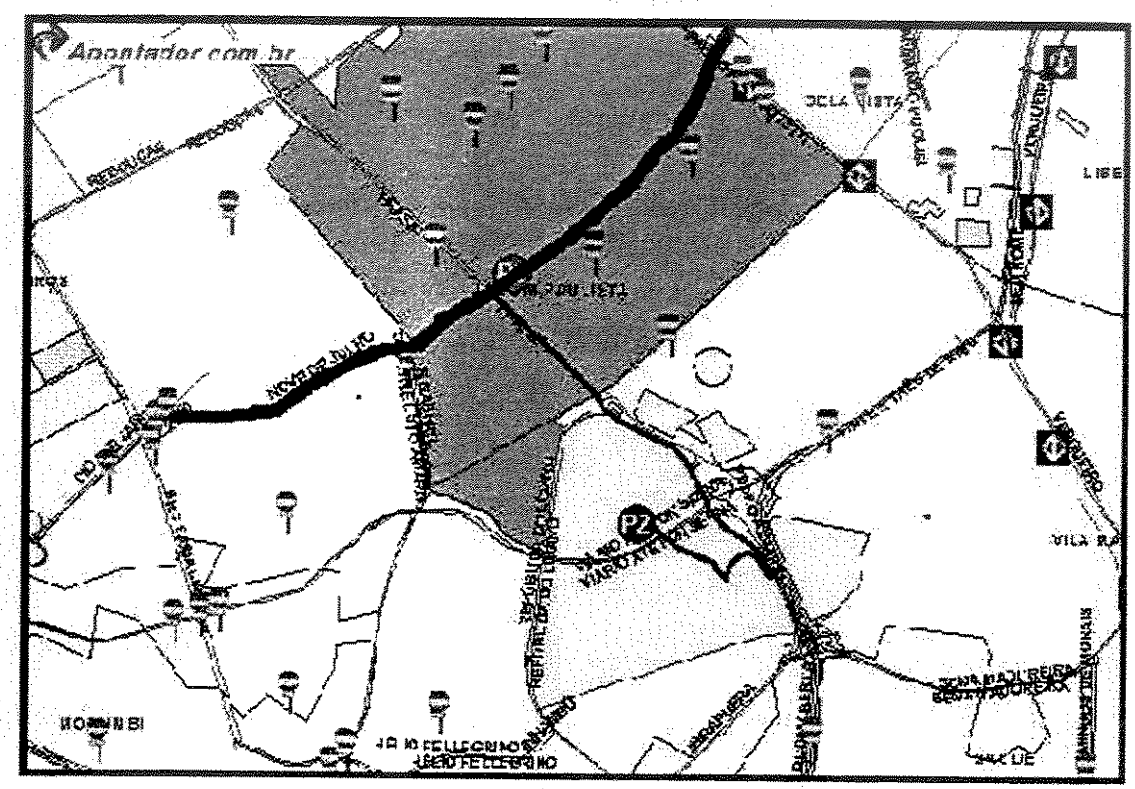


Figura 5.12: Permissão "Ana pode acessar o bairro Jardim Paulista".

Por exemplo, considere a seguinte regra "Ana tem acesso de leitura ao bairro Jardim Paulista" (região mais escura na figura 5.12). O objeto Jardim Paulista é identificado pelo id 501 e corresponde a um polígono, ou seja, a granularidade da permissão é polígono.

Assim sendo, a regra armazenada será $< Ana, 501, leitura >$. Isto subentende que Ana tem acesso ao objeto 501 e também a todos os objetos contidos em 501.

Suponha agora que Ana deseje acessar a Avenida 9 de Julho. Entretanto, o problema é que esta avenida também percorre o bairro Bela Vista e possui apenas uma parte no bairro Jardim Paulista. Neste caso, além da permissão global de acesso ao polígono Jardim Paulista, poderia se introduzir uma permissão com granularidade de linha no sistema para indicar explicitamente que Ana pode acessar toda esta avenida.

Um exemplo de regra usando uma consulta e granularidade de pontos seria quando se deseja permitir que João acesse apenas as estações de metrô (considerando-se que estações de metrô estão representadas como pontos) da Av. Vergueiro. Neste caso, definimos uma permissão (João, todas as estações de metrô da Av. Vergueiro, leitura), onde "todas as estações de metrô da Av. Vergueiro" pode ser especificada como uma consulta em *SQL* espacial. Como se verá ao final deste capítulo, esta consulta será calculada quando houver pedidos de acesso envolvendo o mesmo sujeito.

Este tipo de solução permite o compromisso de gerenciamento de objetos específicos (regras $< s, o, m >$) versus versatilidade em definir autorizações (regras $< s, C, m >$). Resta ainda definir como armazenar e gerenciar as regras, discutido em 5.4.

5.3.2 Especificação do sujeito, objeto e modo de acesso

Sujeitos em um modelo de autorização representam entidades tentando acessar o banco de dados. Como apresentado no capítulo 3, o modelo para dados espaciais não apresenta diferenças quanto ao tipo de usuário, isto é, estes podem ser definidos da mesma forma que nos bancos de dados convencionais. Assim, este modelo considera que sujeitos sejam usuários finais, apenas para facilitar a compreensão. Entretanto, isso pode ser facilmente estendido para incluir papéis ou grupos.

Também para simplificar o modelo, considera-se apenas a leitura e a escrita como os tipos ou modos de acesso padrão. A leitura pode também incluir a visualização de imagens/mapas. Escrita subentende-se permissão para alterar os dados. Dependendo do tipo de aplicação, os tipos de acesso também podem ser facilmente estendidos para incluir novas operações.

O objeto a ser acessado pode ser um componente espacial ou conjuntos deste. Um componente espacial pode ser um polígono, ponto ou linha. Eles podem ser representados diretamente, através de seus identificadores, ou como resultado das consultas.

5.3.3 Especificação das regras de autorização

Definição 3: (Regra de Autorização) Seja s o sujeito, o a especificação do conjunto de objetos autorizados e m o modo de acesso. Uma regra de autorização é definida pela

tripla $< s, o, m >$. Então, de acordo com as definições acima, uma regra de autorização é composta pelos seguintes componentes:

- s: sujeito autorizado;
- o: conjunto de objetos autorizados. Seus elementos são representados por seus identificadores ou por uma consulta. Os tipos de consulta e objetos passíveis de autorização estão no capítulo 4;
- m: modo de acesso, neste caso, pode ser leitura ou escrita.

Segue-se um exemplo de regra de autorização utilizando as consultas geográficas caracterizadas no capítulo 4.

Exemplo (regra do tipo $< s, C, m >$): R: (s: Claudia, o: supermercados que distam menos de 1000 m do Supermercado Barão, m: leitura)

Esta regra de autorização especifica que Claudia tem permissão de leitura a todos os supermercados a menos de 1000 m do Supermercado Barão. Ela pode ser decomposta em:

C: selecione todos os supermercados a menos de 1000 m do Supermercado Barão
o: conjuntos de pontos (supermercados)
predicado métrico: distância (o, ponto específico)

5.3.4 Conjunto de políticas para gerenciar e administrar autorizações

O modelo proposto pressupõe uma administração centralizada de autorizações: apenas o administrador pode conceder e revogar permissões. Com isto, não é necessário pensar em revogação de autorizações em cascata ou não-cascata, como acontece nos modelos convencionais baseados no modelo de controle de acesso seletivo de Griffths e Wade [GW76].

A princípio este modelo não pressupõe a existência de autorizações negativas. Isso deve ser avaliado de acordo com a aplicação em que será utilizado, já que introduz uma complexidade maior nos algoritmos que avaliam um pedido de acesso. Se o mecanismo permitir autorizações negativas, dado um pedido de acesso, é preciso verificar a existência de alguma autorização negativa impedindo o uso deste objeto, antes de permitir o acesso.

Uma vantagem dos sistemas abertos, onde tudo é permitido a menos que exista uma autorização negativa impedindo o acesso, é que em determinados casos o número de autorizações pode ser menor do que nos sistemas fechados, aumentando assim o desempenho

para validar um pedido de acesso. Por exemplo, deseja-se autorizar um sujeito s a acessar todas as cidades do estado de São Paulo, exceto Campinas. Neste caso, é necessário ter apenas uma autorização negativa impedindo o acesso do sujeito s à cidade de Campinas $< s, Campinas, m, - >$. Enquanto que no sistema fechado seria necessário ter autorizações do usuário s a todas as cidades, exceto Campinas.

Uma forma de melhorar o desempenho do mecanismo seria usar autorizações negativas para definir o acesso de leitura dos dados armazenados e autorizações positivas para escrita. Isto é, permitir que tudo possa ser lido a menos que exista uma autorização negativa e permitir a escrita somente se existir uma autorização positiva possibilitando o acesso.

As autorizações espaciais devem ser avaliadas de acordo com a hierarquia decrescente dos objetos definida no capítulo 4: polígono, linha e ponto. Ou seja, se o usuário tem acesso a um polígono, ele tem também às linhas e pontos contidos no polígono. Se tem a uma linha, tem também aos pontos sobre esta linha.

5.4 Armazenamento de regras de autorização

Uma vez definido o modelo, é preciso determinar como ele deve ser implementado. Este é outro aspecto inovador desta dissertação, já que este tipo de consideração não aparece na literatura correlata. O problema pode ser formulado da seguinte forma:

Dado um conjunto de regras de autorização $< s, o, m >$, o que deve ser armazenado junto aos dados no banco de dados para gerenciar o direito de acesso?

A principal questão a levantar diz respeito às vantagens e desvantagens de armazenar permissões junto a cada objeto versus calculá-las dinamicamente.

O armazenamento explícito junto a cada objeto do banco de dados agiliza a verificação de acesso, mas por outro lado traz sérios problemas de espaço em disco e atualização do conjunto de regras. A verificação dinâmica de direito de acesso evita estes problemas, mas aumenta o tempo do processamento da autorização.

Esta seção analisa algumas alternativas, definindo ao final a solução adotada. Estas alternativas se baseiam em decidir se - e quando - armazenar regras $< s, o, m >$ e $< s, C, m >$.

Existem as seguintes alternativas para armazenamento de autorizações:

1. associar a cada objeto do banco de dados os pares $< s, m >$ correspondentes;

2. armazenar regras em separado.

A opção (1) pode tornar-se inviável quando houver uma grande quantidade de sujeitos para um mesmo objeto. Por exemplo, pode haver 1000 técnicos que tenham permissão de acesso a um poste. Neste caso, o objeto poste teria 1000 pares $< s, m >$ associados. Além disso, pode exigir consultar todo o banco de dados a cada atualização de regras.

A opção (2) tem as seguinte possibilidades:

1. armazenar $< s, o, m >$ e $< s, C, m >$ e calcular todas as consultas dinamicamente a cada pedido de acesso;

2. calcular os identificadores de todos os objetos referenciados por consultas $< s, C, m >$ e armazenar apenas $< s, o, m >$;

3. armazenar $< s, o, m >$ e $< s, C, m >$ e, além disso, calcular todas as consultas e armazenar as tuplas $< s, o, m >$, cujo conjunto seria atualizado periodicamente, a critério do administrador.

Em todos os casos, a base de regras precisa ser atualizada quando há modificações no banco de dados. A opção 2.1 é mais flexível, porém é mais cara do ponto de vista de acesso, já que cada pedido de acesso exige calcular ao menos uma consulta e verificar intersecções espaciais. É preferível nos casos em que há muitas atualizações no banco de dados.

Já a opção 2.2 é menos flexível, porém mais barata do ponto de vista de desempenho de acesso. É preferível se há poucas atualizações. Esta alternativa pressupõe que cada regra $< s, C, m >$ seja transformada em um conjunto de regras $< s, o_i, m >$, onde os o_i são o resultado da execução de C no banco de dados. Um grande problema em se armazenar as autorizações $< s, o, m >$ uma a uma é que isto exige o armazenamento de um grande número de permissões, e por conseguinte muito espaço em disco. Além do mais, requer atualização constante do conjunto de regras à medida que o banco de dados é modificado. Isso pode tornar inviável uma aplicação que necessite controlar permissões em um nível de granularidade de pontos. O cálculo das consultas apenas uma vez torna o mecanismo inflexível, ou seja, não se pode atualizar uma permissão especificada através de uma consulta, pois a consulta não foi armazenada, não atendendo as necessidades da grande maioria dos aplicativos atualmente existentes. Por exemplo, se a permissão era "Claudia pode alterar todos os restaurantes do distrito de Barão Geraldo" e o administrador quiser modificar para "todos os supermercados", a consulta especificada já foi perdida. Embora esta alternativa aumente a eficiência para validar um pedido de acesso, perde-se em flexibilidade e consistência dos dados, pois os dados podem estar desatualizados.

Uma opção de compromisso seria 2.3. Neste caso, ganha-se em eficiência, pois as consultas são pré-calculadas no momento de sua inserção. Porém, isso não garante com-

pletamente a consistência do conjunto. Assim, recai-se no problema da eficiência versus consistência.

O modelo propõe utilizar a opção 2.1, levando-se em consideração os requisitos de consistência versus eficiência, sendo que o custo de processar um pedido de acesso neste caso será maior.

5.5 Arquitetura do sistema

Esta seção propõe uma arquitetura básica para o controle de acesso visando apenas a etapa (1) do processo de autorização, ou seja, a verificação do direito a acesso. Pode ser definida pelos seguintes componentes, e está representada na figura 5.13.

- Gerenciador de Autorizações;
- Gerenciador de Controle de Acesso;
- Gerenciador de Consultas;
- Gerenciador de Dados Geográficos.

O Gerenciador de Autorizações é responsável por todo o gerenciamento da base de regras de autorização e possui uma interface para o administrador de segurança. Através desta, o administrador pode inserir, alterar e remover regras de autorização. O Gerenciador de Controle de Acesso implementa o algoritmo de controle de acesso 5.4 a ser apresentado mais adiante na seção 5.6. O Gerenciador de Consultas processa tanto as consultas dos pedidos de acesso quanto as consultas das regras de autorização e retorna os objetos o solicitados na consulta. O Gerenciador de Dados Geográficos é responsável por executar as consultas geográficas e manipular os objetos espaciais.

O administrador de segurança é o único usuário que tem permissão para inserir novas autorizações. O administrador deve especificar o sujeito, o modo de acesso e o objeto através de seu identificador ou consultas.

O Gerenciador de Autorizações deve prover uma interface que facilite a identificação de objetos espaciais, caso o usuário queira definir a regra de autorização usando o identificador do objeto. Por exemplo, caso ele queira definir uma autorização para uma cidade, a interface deve exibir uma lista com os identificadores e os nomes das cidades disponíveis para acesso. Deve também prover uma interface para que se possa especificar as consultas sem grandes complicações. Estas regras são armazenadas na base de regras de autorização. Consultas devem ser armazenadas usando SQL espacial.

Quando um usuário ou aplicativo faz um pedido de acesso PA, o Gerenciador de Controle de Acesso valida ou não a autorização face ao pedido PA. Para isso, o Gerenciador

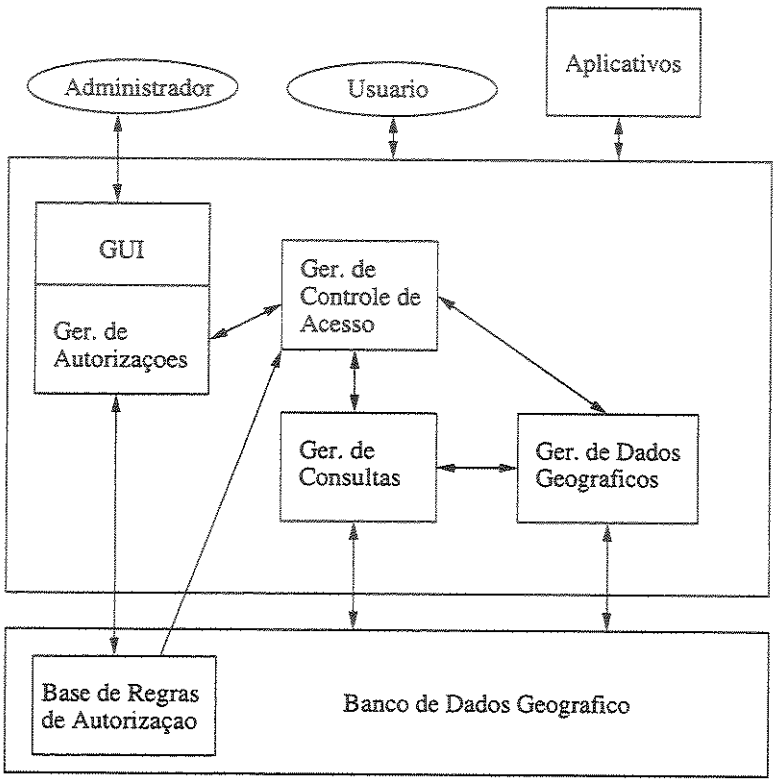


Figura 5.13: Arquitetura do modelo.

executa o algoritmo 5.4 consultando a base de regras de autorização. Caso necessário, comunica-se com o Gerenciador de Consultas a fim de processar a consulta do pedido de acesso PA e a consulta armazenada na regra de autorização.

O Gerenciador de Consultas calcula a consulta e recupera os elementos correspondentes. Ressalta-se que, como visto no capítulo 4, a consulta não é obrigatoriamente espacial. Neste caso, não há necessidade de comunicação com o Gerenciador de Dados Geográficos. Caso a consulta envolva predicados geográficos, é enviada ao Gerenciador de Dados Geográficos, que retorna os elementos espaciais resultantes para o Gerenciador de Consultas. Este, por sua vez, recebe os elementos, já com os seus identificadores e os devolve para o Gerenciador de Controle de Acesso.

O Gerenciador de Controle de Acesso tem necessidade de comunicação com o Gerenciador de Autorizações caso seja necessário modificar a base de regras para garantir o direito a acesso. Estes são os casos de implementar, por exemplo, a segunda opção dos casos polígono-polígono e polígono-linha da seção 5.2.1. Neste cenário, o administrador é avisado e deve inserir explicitamente novas regras de autorização.

5.6 Algoritmos para gerenciar regras e analisar pedidos de acesso

O controle de acesso exige dois tipos de algoritmos - atualização da base de regras (figura 5.13) e leitura desta base quando há pedidos de acesso. Os algoritmos 5.1, 5.2 e 5.3 permitem gerenciar a base de regras e o algoritmo 5.4 permite validar um pedido de acesso.

No modelo proposto, considera-se que os conflitos segundo a Definição 1 da seção 5.1.1 são resolvidos no momento da inserção de regras e não no momento da validação de pedidos de acesso. Cabe ao administrador decidir se irá permitir regras, mesmo que estas tenham conflitos segundo a Definição 1. Portanto o algoritmo 5.1 verifica os conflitos do tipo 1.

Considerando-se que todas as regras da base de regras estejam consistentes de acordo com a política definida pelo administrador, o algoritmo de validação dos pedidos de acesso 5.4 verifica apenas os conflitos segundo a Definição 2. O direito de acesso é concedido somente se houver uma regra explícita autorizando o sujeito a acessar aquele objeto com aquele tipo de modo de acesso. Ou se o objeto requerido estiver contido dentro de algum outro objeto a que o usuário possui uma autorização explícita.

Algoritmo 5.1 Inserção de regras
Hipótese: Pressupõe-se que a inserção não tornará a base inconsistente.
Input: pedido de inserção de regra $R =< s, Q, m >$, onde Q é uma consulta que define o conjunto de objetos da regra.
Output: base de regras atualizada.

1. Transformar Q em consulta C, em SQL ou SQL espacial e executá-la.
2. Seja $O = o_i$, o conjunto de objetos retornados pela consulta C. Para cada $o_i \in O$, verificar se há conflito com alguma regra existente de acordo com a Definição 1 de 5.1.1.
3. Se O contém somente um objeto e houver conflito, solicitar decisão do administrador sobre inserção ou não de $< s, O, m >$.
4. Se O contém somente um objeto e não houver conflito, insere $< s, O, m >$.
5. Se O contém mais de um objeto e houver conflito com algum o_i, solicitar decisão do administrador sobre inserção ou não de $< s, C, m >$.
6. Se O contém mais de um objeto e não houver conflito algum, insere $< s, C, m >$.

Algoritmo 5.2 Eliminação de regras
Input: pedido de eliminação de regra $R =< s, Q, m >$.
Output: base de regras atualizada.

1. Transformar Q em consulta C, em SQL ou SQL espacial e executá-la.
2. Seja $O = o_i$ o conjunto de objetos retornados pela consulta C.
3. Recuperar todas as regras $< s, o, m >$ e $< s, C, m >$, tais que $o \bigcap o_i \neq 0$ e $C \bigcap o_i \neq 0$.
4. Apresentar estas regras ao administrador, que decidirá quais efetivamente eliminar.

Algoritmo 5.3 Modificação de regras

Pressupõe-se que toda modificação será realizada através de combinação de remoção (algoritmo 5.2) e inserção (algoritmo 5.1) de regras.

Algoritmo 5.4 Validação de um pedido de acesso

Input:

[1] pedido de acesso (S, C_a, M).

[2] conjunto de regras de autorização (s, o, m) e (s, C, m) existentes na base de regras.

Output:

[1] AUTORIZADO.

[2] NEGADO (possivelmente envolvendo alerta ao administrador).

1. Ao receber um pedido de acesso $PA =< S, C_a, M >$ representado pela consulta C_a, selecione regras de autorização $r_i =< s, o, m >$ e $r_j =< s, C, m >$ da base de regras tais que $s = S$ e $m = M$. O resultado desta etapa é um conjunto de regras $RA = < S, o_i, M > \bigcup < S, C_i, M >$.
2. Calcule as consultas das regras $< S, C_i, M >$ para determinar os objetos referenciados, obtendo o conjunto final de regras $RF = < S, o_k, M >$.
3. Calcule a consulta C_a, obtendo $PA = < S, o_a, M >$, que determina os objetos atingidos pelo pedido de acesso.
4. Detectar problemas relativos aos conflitos segundo Definição 2 entre PA e os objetos de RF.
5. Resolva conflitos segundo políticas definidas em 5.2.2 garantindo ou não o direito de acesso.

Detalhando melhor os passos (4) Detectar conflitos e (5) Resolver conflitos, temos o seguinte algoritmo:

1. Se $PA \subset RF$ então acesso CONCEDIDO. (Como todos os oids dos objetos já foram calculados nos passos 2 e 3 do algoritmo 5.4, trata-se apenas de uma operação padrão de junção em banco de dados.)
2. Senão, para cada $r_a =< S, o_a, M > \in PA - RF$ verificar se o_a está contido em algum o_k. Se estiver, o acesso a o_a é concedido.

3. Se o_a engloba totalmente algum o_k, o acesso é negado, pois o_a é que deveria estar dentro de o_k.

4. Se houver pedidos ainda não resolvidos para objetos o_a, existem conflitos de intersecção parcial que devem ser resolvidos segundo políticas da seção 5.2.2.2.

O algoritmo 5.4 funciona da seguinte maneira. Ao receber um pedido de acesso, seleciona todas as regras de autorização contidas na base de regras com o mesmo sujeito e modo de acesso. Isto é importante no algoritmo, porque várias requisições já podem ser descartadas neste momento, sem perder tempo em calcular as consultas e recuperar os objetos.

No modelo proposto, um sujeito pode ter várias autorizações. Portanto, podem haver várias regras de autorização $< s, o, m >$ e $< s, C, m >$ resultantes da consulta anterior. Em seguida as consultas armazenadas nas regras $< s, C, m >$ e do pedido de acesso são calculadas. Então é só determinar se todos os pedidos de acesso possuem uma regra de autorização equivalente. Se sim, o acesso é concedido. Caso contrário, verifica-se se cada objeto do pedido de acesso (que não tiver uma regra equivalente) está contido em algum objeto de alguma regra armazenada. Se todos estiverem, o acesso é concedido. Se houver objetos parcialmente contidos, então os conflitos devem ser resolvidos segundo as políticas da seção 5.2.2.2 .

Seja o seguinte exemplo para ilustrar o funcionamento do algoritmo 5.4.

Base de regras:

$r_1 =< s, Campinas, m >$

$r_2 =< s,$ *"todos os supermercados de Paulínia"*$, m >$

$r_3 =< s_2,$ *Jardim Santa Genebra*$, m >$

Caso 1: $PA =< s, Campinas, m >$

As regras r_1 e r_2 são recuperadas, a consulta em r_2 é executada e obtemos

$RF = r_1 \bigcup < s, supermercado1, m >, ... < s, supermercadoN, m >$.

Como $PA \subset RF$, acesso concedido pelo passo 4.1.

Caso 2: $PA =< s,$ *Cambuí*$, m >$

Neste caso, como Cambuí está contido em Campinas, o acesso concedido pelo passo 4.2.

Caso 3: $PA =< s,$ *Paulínia*$, m >$

Neste caso, embora Paulínia englobe os supermercados ela não está contida em supermercados, então o acesso é negado pelo passo 4.3.

Caso 4: $PA =< s, RodoviaD.Pedro, m >$

Neste caso, cai-se no passo 4.4, que corresponde à situação de intersecção parcial polígono-linha. Este caso tanto pode redundar em permissão de acesso quanto decomposição da rodovia em trechos menores e permissão parcial apenas para o trecho contido em Campinas.

Caso 5: $PA =< s_3, Americana, m >$

Acesso negado pelo passo 1.

5.7 Arquitetura alternativa

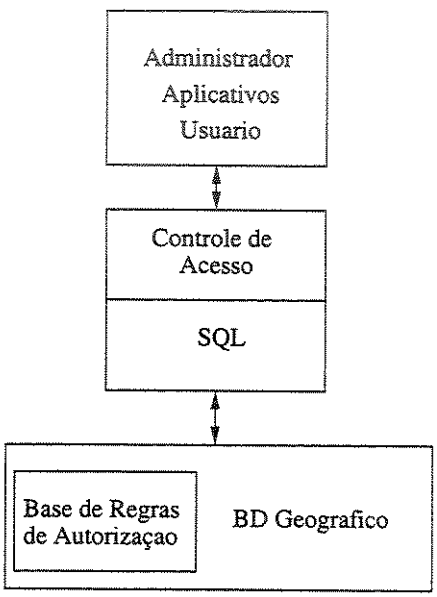


Figura 5.14: Arquitetura alternativa.

A arquitetura proposta na seção 5.5 exige verificação de controle de acesso em vários níveis. Uma arquitetura alternativa, mais simples, está representada na figura 5.14. Esta alternativa propõe que o mecanismo de controle de acesso geográfico seja implementado diretamente como uma camada sobre o processamento de consultas *SQL*. Isto evita acesso direto aos dados via *SQL*. Neste caso, temos apenas um Gerenciador de Controle de Acesso.

Este tipo de abordagem é importante a fim de garantir a segurança do sistema, já que a implementação do controle de acesso no nível de aplicação (seção 5.5) deixa um brecha

de segurança. Ele permite que o usuário acesse os dados usando *SQL* diretamente. Esta proposta alternativa evita acessos não permitidos, mas por outro lado, é menos flexível que a anterior.

5.8 Conclusão

Esta seção apresentou o modelo de controle de direito a acesso proposto na dissertação. O modelo é baseado na definição de regras $< s, o, m >$, armazenados de acordo com a granularidade de o em $< s, o, m >$ ou $< s, C, m >$. O capítulo propõe uma arquitetura para o sistema e define algoritmos de gerência de regras e gerência de controle de acesso. O próximo capítulo utiliza o modelo aqui proposto adaptando-o para uma aplicação real no sistema SAGRE.

Capítulo 6

Controle de acesso no sistema SAGRE

O SAGRE controla e opera automaticamente o Mapa Urbano, o estudo de mercado e demanda, o registro das redes de canalização e cabos (ópticos e metálicos), o projeto de engenharia e manutenção, o processo de planejamento empresarial e a gerência das facilidades e manutenção da rede.

O SAGRE é utilizado em diversos setores da empresa operadora e pessoas com diferentes perfis devem ter acesso aos dados. Dessa forma, constatou-se uma grande necessidade em se controlar o acesso tanto aos aplicativos quanto às operações executadas nesses aplicativos. Além disso em alguns aplicativos, constatou-se também a necessidade de se ter um controle de acesso por áreas geográficas, já que o SAGRE permite a manipulação de dados geográficos.

Este capítulo pretende apresentar o sistema SAGRE de uma forma geral focando em suas necessidades de controle de acesso e mostrando como a proposta da dissertação pode ser adotada no SAGRE. O capítulo está organizado da seguinte forma. A seção 6.1 apresenta o sistema SAGRE. A seção 6.2 a sua aquitetura. A seção 6.3 apresenta o controle de acesso existente atualmente. A seção 6.4 apresenta as necessidades de controle de acesso geográfico. E finalmente, a seção 6.5 apresenta uma breve conclusão do capítulo.

6.1 Apresentação do SAGRE

O SAGRE é um Sistema Automatizado de Gerência de Rede Externa que consiste de um conjunto de aplicativos com objetivo de automatizar os processos relacionados à rede externa. Ele visa atingir os seguintes objetivos:

- Facilitar o processo de planejamento: decisões podem ser tomadas levando em con-

sideração a rede de telecomunicações existente, informações de demanda, política de telecomunicações e regras de engenharia e operação.

- Automatizar o processo de projeto e estimativa de custos: novos projetos são realizados refletindo regras de engenharia e restrições aplicáveis. Eles são gerados com o uso de ferramentas de projeto automatizadas. O processo também utiliza informações sobre o modelo de construção, permitindo a estimativa de custos de materiais e mão-de-obra. Isto torna possível a avaliação do custo de projetos alternativos e a geração automática da lista de materiais e mão-de-obra necessários.

- Maior eficácia na implantação de projetos: o SAGRE reduz o tempo de projeto, construção e implantação da rede, disponibilizando-a rapidamente para a operação.

- Gerência eficiente da ocupação da rede: através da automação dos processos de gerência de facilidades é possível controlar de forma precisa a ocupação das facilidades, sua disponibilização nas áreas de atuação da empresa operadora, os níveis de saturação da rede, auxiliando na tomada de decisão para futuras expansões e planejamentos mais adequados.

- Gerência e visibilidade corporativa da rede: através das funções de manutenção do cadastro, projeto, demanda, operação do SAGRE fica garantido que os registros da planta externa reflitam com precisão a rede que está nas ruas, cobrindo todo o ciclo de vida e processos da rede externa. Para isto, o banco de dados do SAGRE inclui o mapeamento urbano, a localização dos clientes, as estruturas de suporte e todos os elementos da rede, informações de demanda e ocupação.

Para tal, o SAGRE compõe-se de vários módulos. Entre eles podemos citar: SAGRE/ Adm, SAGRE/Cad, SAGRE/Conv, SAGRE/Oper, SAGRE/FCT, SAGRE/Conv-Oper, SAGRE/Market, SAGRE/Tup, SAGRE/Viewer e SAGRE/Relatórios. Dentre estes, os módulos importantes para o estudo do controle de acesso são SAGRE/Adm, SAGRE/Cad e SAGRE/Tup.

6.1.1 SAGRE/Adm

O SAGRE/Adm tem como propósito facilitar tarefas que devem ser realizadas pelo administrador do SAGRE na operadora. Entre elas estão as seguintes atividades:

- Segurança: permite gerenciar os usuários, cadastrando usuários e permissões;

- Gerência dos dados: permite criar e remover bancos de dados do SAGRE, alterar senhas destes bancos e realizar backups;

- Configuração: permite adequar o SAGRE às necessidades de cada empresa operadora, pois as diversas empresas que usam o SAGRE trabalham de forma diferente em vários aspectos tanto no modo de realizar o cadastro, projeto e operação da rede externa quanto no modo de operar com o próprio sistema SAGRE. Um exemplo são as configurações de material e mão-de-obra utilizados nos projetos realizados pelas empresas operadoras.

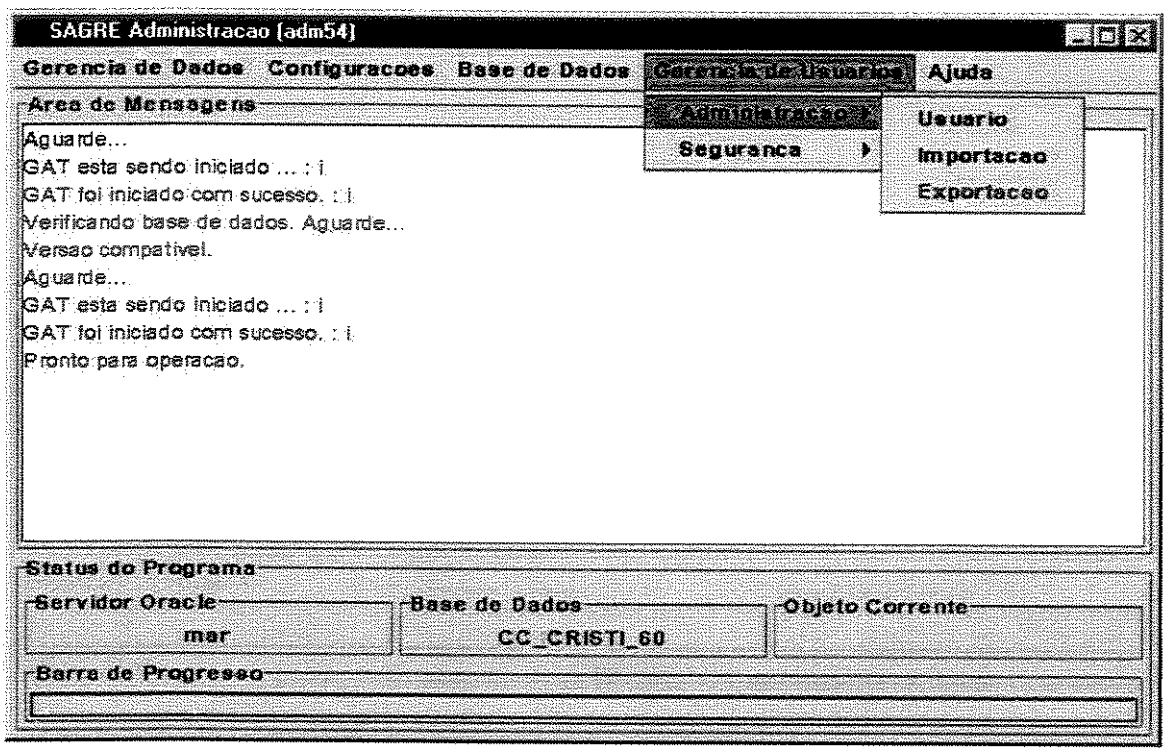


Figura 6.1: SAGRE/Adm.

A figura 6.1 mostra uma cópia de tela do SAGRE/Adm, enfatizando a disponibilização de funções de segurança.

6.1.2 SAGRE/Cad

Um conceito fundamental na modelagem de sistemas para aplicações de concessionárias de serviços públicos é o MUB - Mapa Urbano Básico [CCH+96]. Este modela os elementos que formam a base do planejamento urbano. Em princípio, todas as aplicações relativas a serviços utilitários (telefonia, eletricidade e outros) usam o MUB como base, porém

os elementos utilizados variam conforme a aplicação. Exemplos de elementos que fazem parte do MUB são divisas de lote, imóveis (numeração predial e edificações de destaque).

No caso do SAGRE, os elementos do MUB incluem: informações sobre localidade, arruamento (quadra, meio fio, logradouro), loteamento (indicação de lote, numeração), aspectos geográficos (hidrografia, acidentes geográficos) e obras públicas (rodovia, ferrovia, edificações de destaque como escolas, igrejas).

O módulo SAGRE/Cad mantém o cadastro do MUB e da rede externa de telecomunicações. Na rede externa estão presentes as informações de infra-estrutura utilizadas pelos serviços de telecomunicações, compreendendo desde a estação telefônica (ET), rede de canalização (rede de dutos), rede subterrânea (esquemas de emendas e caixas subterrâneas) e rede aérea (postes, cabos aéreos, armários de distribuição, caixas terminais). Este módulo permite elaborar projetos novos ou de expansão.

Um exemplo de uma tela com objetos de MUB e rede é mostrado na figura 6.2.

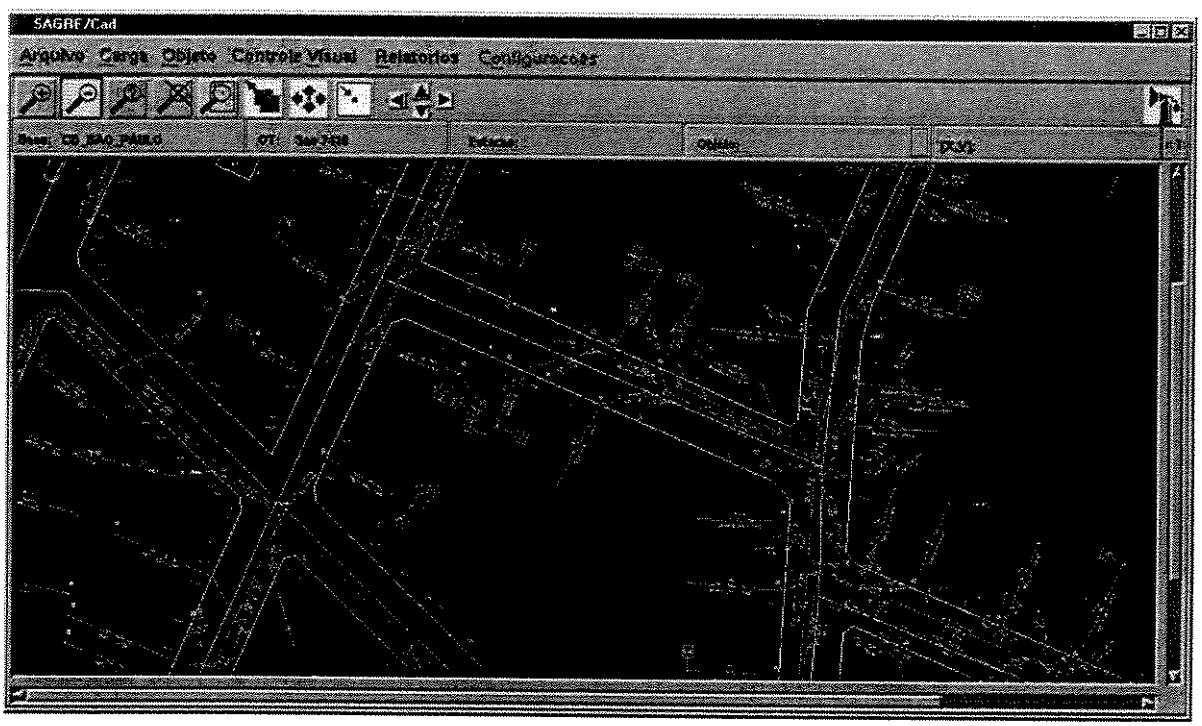


Figura 6.2: SAGRE/Cad.

As alterações sobre a rede externa de telecomunicações se dão por meio de ampliações e expansões. As alterações são denominadas de "projeto". A criação de projetos requer a identificação de um responsável e a área de atuação do trabalho em forma de coorde-

nadas geográficas, definidas como um polígono. Os projetos são realizados no módulo SAGRE/Cad, sobre o próprio banco de dados de cadastro.

6.1.3 SAGRE/Tup

A Agência Nacional para as Telecomunicações (ANATEL) exige das empresas operadoras o atendimento de índices de qualidade. Um destes índices é o que define o serviço de Telefones de Uso Público (TUP). O SAGRE/Tup permite cadastrar informações de TUPs na base de dados do SAGRE e através do uso de informações levantadas em campo, geoprocessamento e informações de satélite permite que as empresas operadoras de telecomunicações possam verificar se estão atendendo as metas da ANATEL para TUP.

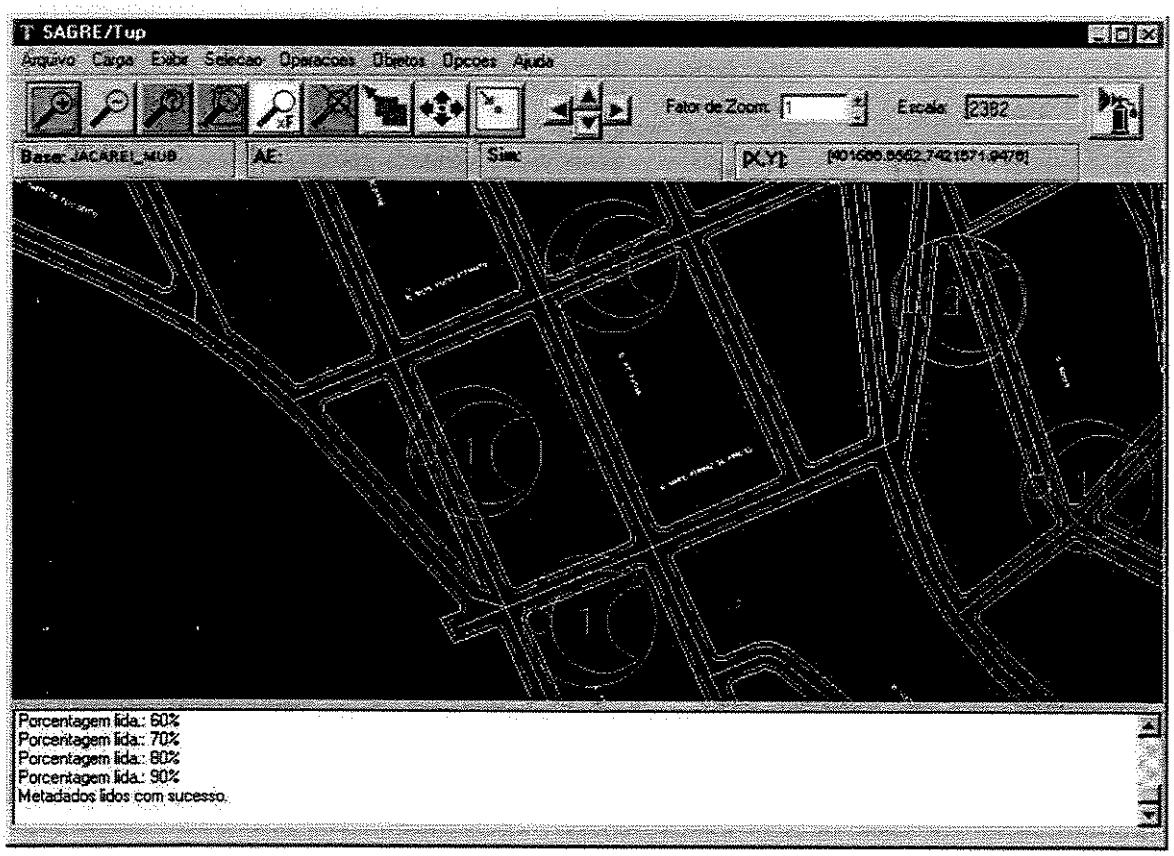


Figura 6.3: Representação de TUPs no SAGRE/Tup.

Por exemplo, a ANATEL determina que um domicílio não pode estar além de 800 metros de um TUP caminhando-se por logradouros. Esta regra é válida para todas as empresas de telefonia fixa originárias do sistema TELEBRÁS em todas as localidades de

sua área de concessão. Esta meta foi alterada para 300 m e deve ser atendida até 2003. As empresas que conseguirem antecipar esta meta poderão atuar na área de celulares. Assim, várias empresas operadoras estão tentando antecipá-la.

O SAGRE/Tup mostra a cobertura dos TUPs através de círculos com raio de tamanho definido pelo usuário. Por inspeção visual, o usuário pode verificar se a área urbana da localidade está coberta pelas áreas de atendimento dos seus TUPs. O SAGRE/Tup também informa o melhor local para instalação de um novo TUP para as áreas não cobertas. A figura 6.4 mostra uma cópia de tela do SAGRE/Tup contendo indicações de áreas a descoberto.

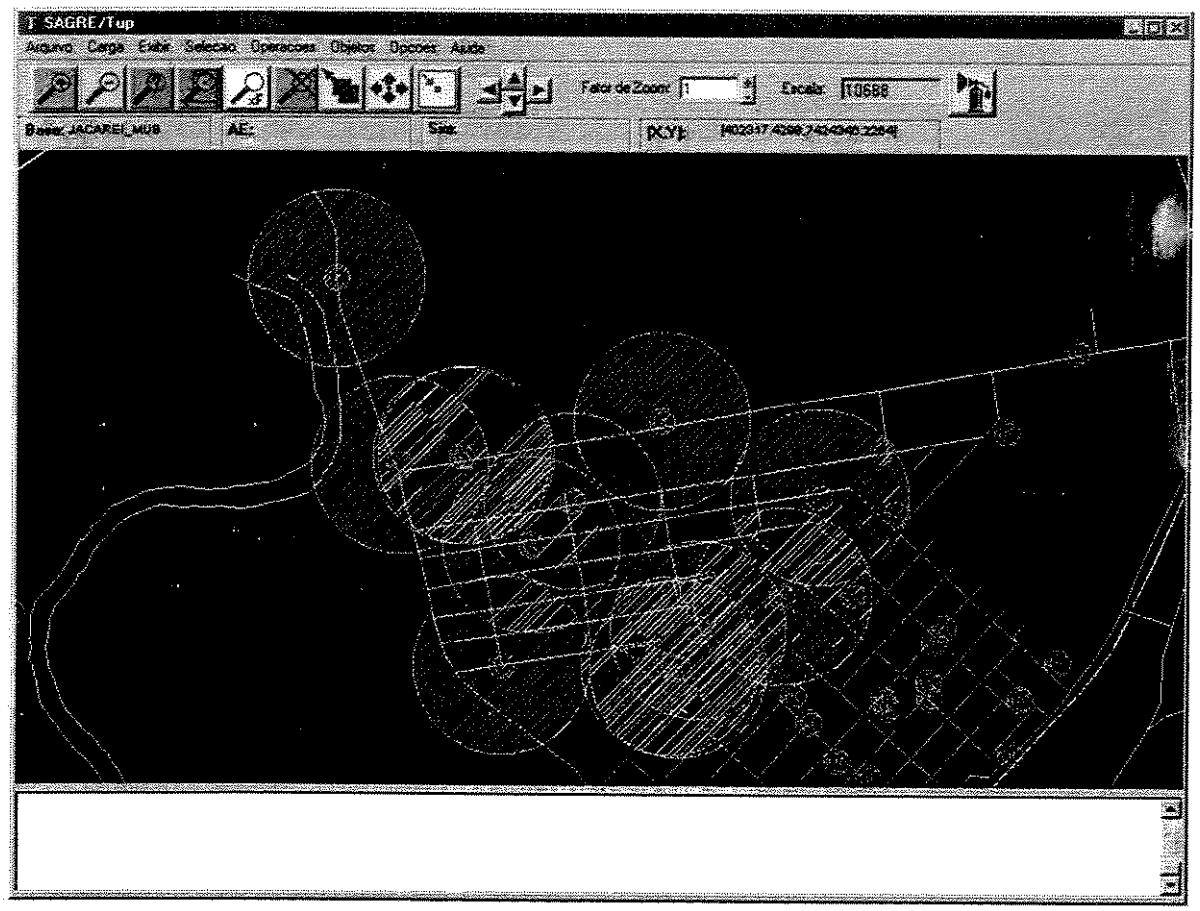


Figura 6.4: Área de cobertura de TUPs.

6.2 Arquitetura do SAGRE

A arquitetura do SAGRE é composta de três camadas funcionais [CCH+96], que executam tarefas bem definidas e funcionalmente diferentes: a Interface Homem-Máquina (IHM), a camada de Aplicação (APL) e a de encapsulamento da Base de Dados (EBD), organizadas conforme a figura 6.5:

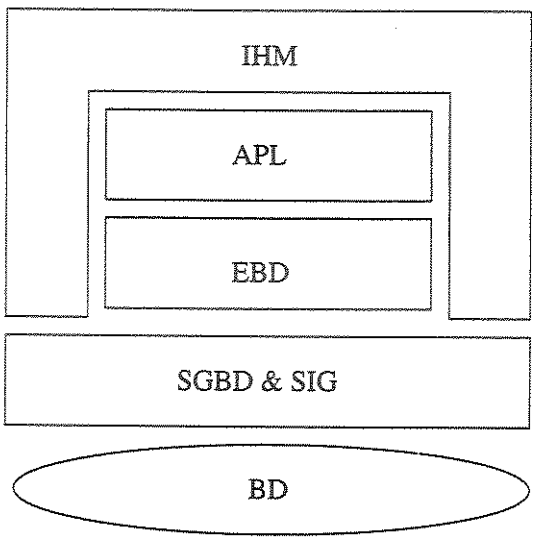


Figura 6.5: Arquitetura do SAGRE.

Essas camadas atuam sobre um banco de dados associado a um sistema de informação geográfico. A camada IHM implementa todos os processos de comunicação com o usuário, sendo responsável pela padronização da interface homem-máquina do SAGRE. Essa interface é baseada em um conjunto de telas cujos conteúdos incluem mapas, atributos e mensagens, com formas distintas de tratamento para cada um desses elementos.

A camada APL provê todas as operações de atualização (inserção, alteração e remoção) e consulta sobre os objetos do banco de dados, centralizando as tarefas da aplicação propriamente dita. Quando o usuário seleciona um objeto de rede externa em uma tela de mapa e escolhe uma operação a ser executada, esta camada é ativada. Ela também é responsável pelos procedimentos operacionais do sistema, tais como garantir a consistência dos dados a partir das regras de engenharia e de gerenciamento.

A camada EBD reúne todas as rotinas de acesso ao banco de dados, sendo responsável pela padronização do seu acesso, pela abstração em relação às tabelas físicas do sistema e segurança na interação dos módulos IHM e APL com a base de dados, o que se dá através

da invocação de primitivas funcionais pré-definidas.

O SAGRE utiliza o SGBD Oracle e o sistema de informação geográfica (SIG) VISION*. O SGBD Oracle provê o ambiente para armazenamento e manipulação dos dados do SAGRE. A representação das informações geográficas é feita pelo VISION*, um produto desenvolvido pela empresa AutoDesk. O VISION* permite a manipulação dos dados geográficos estendendo esta capacidade ao SGBD Oracle.

6.2.1 Gerenciador de Atributos

Um dos componentes da camada de interface com o usuário é o Gerenciador de Atributos, utilizado por quase todos os módulos do SAGRE, incluindo SAGRE/Adm, SAGRE/Cad e SAGRE/Tup. O papel do gerenciador de atributos é de intermediar o acesso do usuário aos dados armazenados no banco de dados, sendo um "mediador" entre interface e a camada APL.

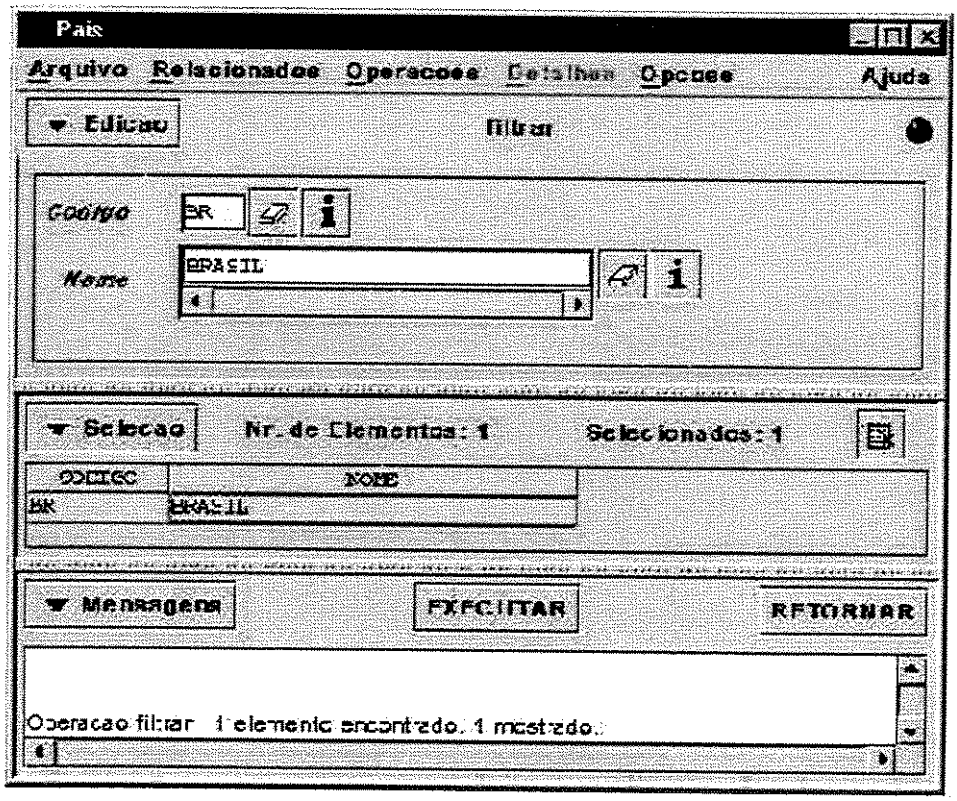


Figura 6.6: Interface de atributos do SAGRE.

A interface deste gerenciador funciona gerando dinamicamente telas customizadas de objetos do SAGRE com todos os seus atributos alfanuméricos. Cada tipo de solicitação do usuário gera dinamicamente uma tela interativa para visualização ou atualização de atributos alfanuméricos. O formato e campos desta tela dependem do tipo de objeto a ser consultado/manipulado e do módulo do SAGRE ativo no momento.

Esta construção dinâmica e contextual de telas do Gerenciador de Atributos é baseada em um conjunto de metadados armazenados no próprio banco de dados. A figura 6.6 apresenta uma cópia de tela do Gerenciador de Atributos exibindo dados sobre um País, neste caso Brasil.

6.2.2 Vision

O VISION* é um produto desenvolvido pela divisão VISION* Solutions da Autodesk. O VISION* faz o gerenciamento e manipulação de dados espaciais estendendo as características do Sistema Gerenciador de Banco de Dados (SGBD) Oracle para tratar esses tipos de dados. Esta característica torna o VISION* muito especial, pois ao contrário da maioria dos SIGs, ele utiliza um SGBD relacional comercial para representar não só os dados alfanuméricos (convencionais) mas também os dados geográficos. Assim, um lance de cabo além de sua representação convencional através de uma tabela de atributos, possui uma representação gráfica cuja geometria também está armazenada em tabelas do mesmo banco de dados.

Na realidade, o VISION* estende as características do SGBD permitindo o armazenamento e recuperação dos dados espaciais (multidimensionais). Em outras palavras, o VISION* coloca-se como uma camada entre o aplicativo e o SGBD. Assim, os aplicativos interagem com o VISION* por meio de seus utilitários, deixando a cargo destes a interação com o SGBD.

6.3 Controle de acesso no SAGRE

6.3.1 Níveis de acesso

Atualmente, o controle de acesso no SAGRE é feito sobre operações dentro dos aplicativos e também sobre a ativação destes aplicativos/módulos. Em outras palavras, o controle é exercido em termos de permissão de operação e não acesso a dados. Usando a terminologia da dissertação, o controle de acesso do SAGRE é do tipo <s, -, m>, onde m neste caso é um conjunto de módulos e operações disponíveis nestes módulos. Este controle é exercido sobre usuários cadastrados no sistema SAGRE através do módulo SAGRE/Adm. O acesso ao módulo SAGRE/Adm também é controlado e somente é permitido aos usuários

administradores (super-usuários) do SAGRE. O administrador do SAGRE deve utilizar o SAGRE/Adm para cadastrar e atribuir permissões a outros usuários do SAGRE. Apenas para ilustrar, a tela do Ativador Unificado é mostrada na figura 6.7.

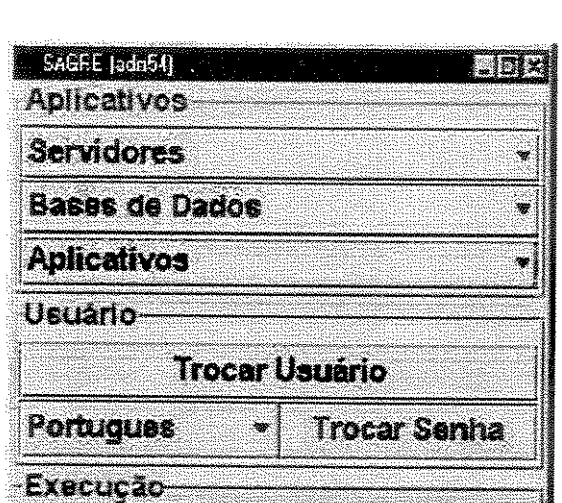


Figura 6.7: Ativador Unificado.

Todos os usuários do sistema SAGRE precisam se identificar e autenticar no Ativador Unificado para poderem utilizar algum módulo. Sempre que um servidor é selecionado ou o botão Usuário é ativado, uma janela de identificação e autenticação, tal qual mostrada na figura 6.8, é aberta. O módulo, o banco e o servidor somente são habilitados para usuários autenticados. A execução de um módulo do SAGRE é sempre associada ao usuário autenticado no Ativador Unificado.

O controle do SAGRE é baseado na noção de perfis, representados por permissões. Essas permissões (também chamadas autorizações) são atribuídas a usuários durante o processo de cadastramento do usuário. O SAGRE proporciona atualmente 9 tipos de permissões pré-configuradas para que os usuários possam utilizar os módulos. Listamos a seguir apenas as que interessam nesta dissertação:

- **atualização:** permite ao usuário ativar o módulo SAGRE/Cad e executar operações neste módulo. As atualizações na rede externa são feitas por meio de projetos. A criação desses projetos é realizada no módulo SAGRE/Cad. Com esta autorização, o usuário pode criar projetos, definir um projeto como corrente e realizar operações sobre ele, desde que o projeto tenha sido criado por ele mesmo;

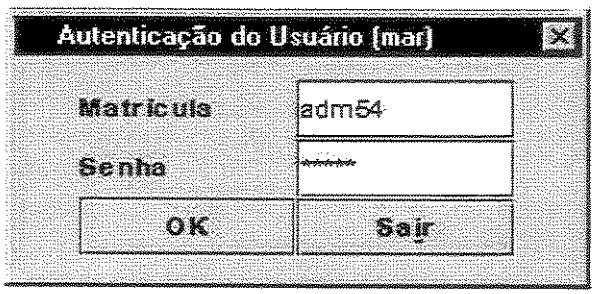


Figura 6.8: Autenticação do usuário.

- **implantação:** permite ao usuário ativar o módulo SAGRE/Cad e executar operações neste módulo. Esta autorização sobrepõe a de atualização permitindo ao usuário criar os seus próprios projetos, além de poder atualizar e implantar projetos de qualquer usuário;

- **abrir banco:** permite ao usuário acessar os módulos do SAGRE apenas para leitura. Os usuários poderão utilizar os módulos sem, entretanto, poder atualizá-lo em qualquer tipo de operação.

Pode-se dizer que as permissões podem variar de acordo com o contexto. No SAGRE/Cad, por exemplo, as permissões variam dependendo do projeto em que o usuário está trabalhando. Um usuário só tem permissão para atualizar ou implantar um projeto que não seja seu se tiver a permissão de "implantação". Se a permissão for "atualização", ele só pode atualizar e implantar projetos de sua autoria.

Como o controle de acesso é feito também sobre a ativação dos módulos, para ativar um determinado módulo, um usuário autenticado deve possuir, no banco e instância Oracle especificados, alguma permissão associada àquele módulo. O SAGRE permite ativar os módulos de interesse desta dissertação de acordo com o seguinte agrupamento de permissões apresentado na tabela 6.1.

Módulo	Permissão
SAGRE/Adm	administrador (super-usuário)
SAGRE/Cad	abrir banco, implantação, atualização
SAGRE/Tup	abrir banco, implantação, atualização

Tabela 6.1: Permissões e módulos.

6.3.2 Gerência de usuários

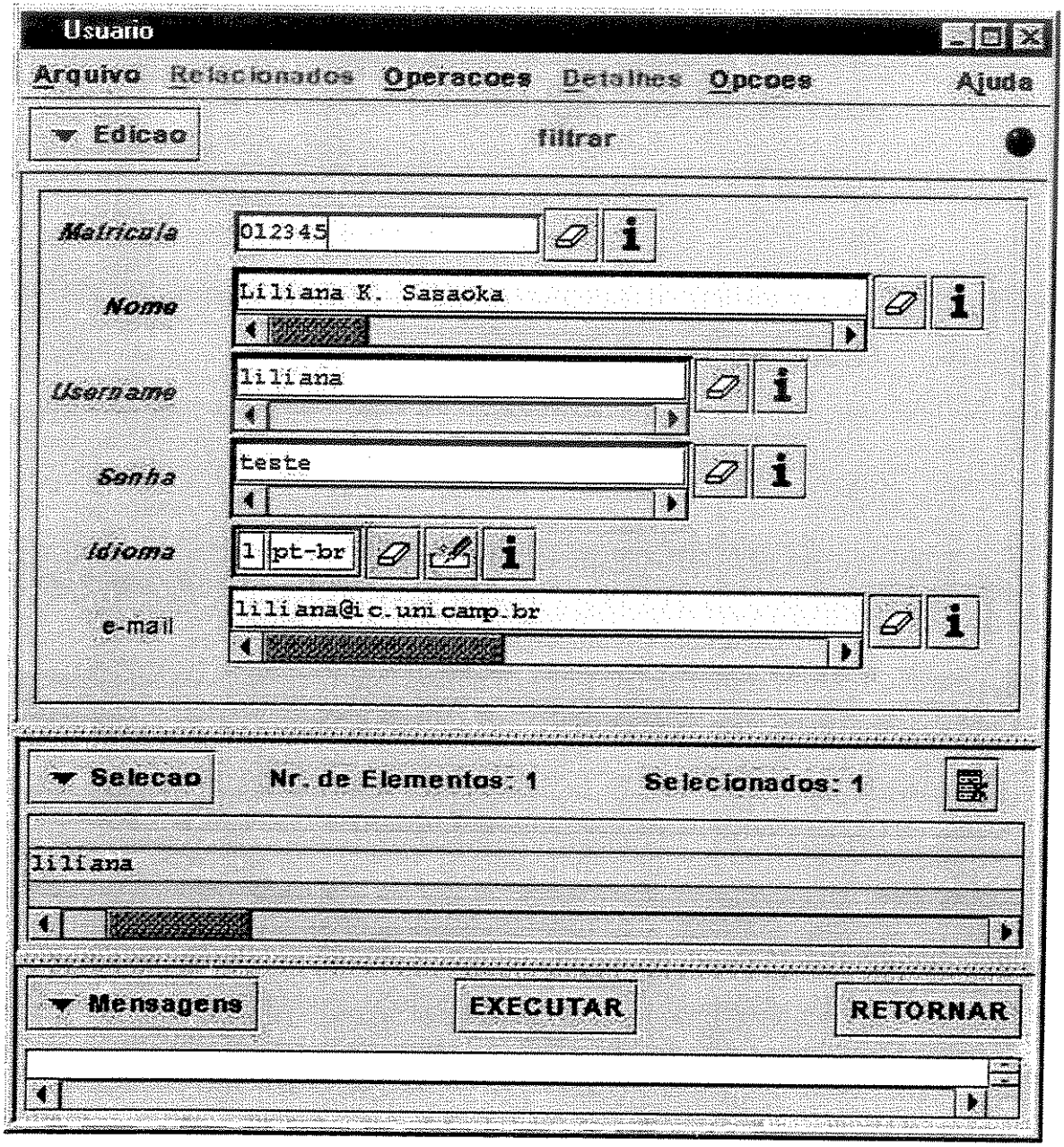


Figura 6.9: Tela de cadastramento de usuário.

O módulo SAGRE/Adm permite gerenciar os usuários com opções de inserção, remoção, alteração e consulta. Os atributos de um usuário são:

- matrícula: identificador único do usuário no SAGRE;
- nome do usuário;
- username: conta unix vinculada a este usuário;
- senha: senha para autenticação do usuário, é criptografada no cadastramento do usuário;
- idioma: idioma de apresentação do SAGRE. O idioma padrão é o português;
- e-mail do usuário.

A figura 6.9 apresenta uma cópia de tela de cadastramento de usuário.

Uma vez que o usuário foi cadastrado, é preciso atribuir-lhe uma permissão. O SAGRE/Adm permite cadastrar apenas um tipo de permissão para um usuário em um determinado banco de dados. O super-usuário pode utilizar as operações de inclusão, remoção e alteração de permissão para um determinado usuário.

A figura 6.10 apresenta uma cópia de tela de atribuição de permissão.

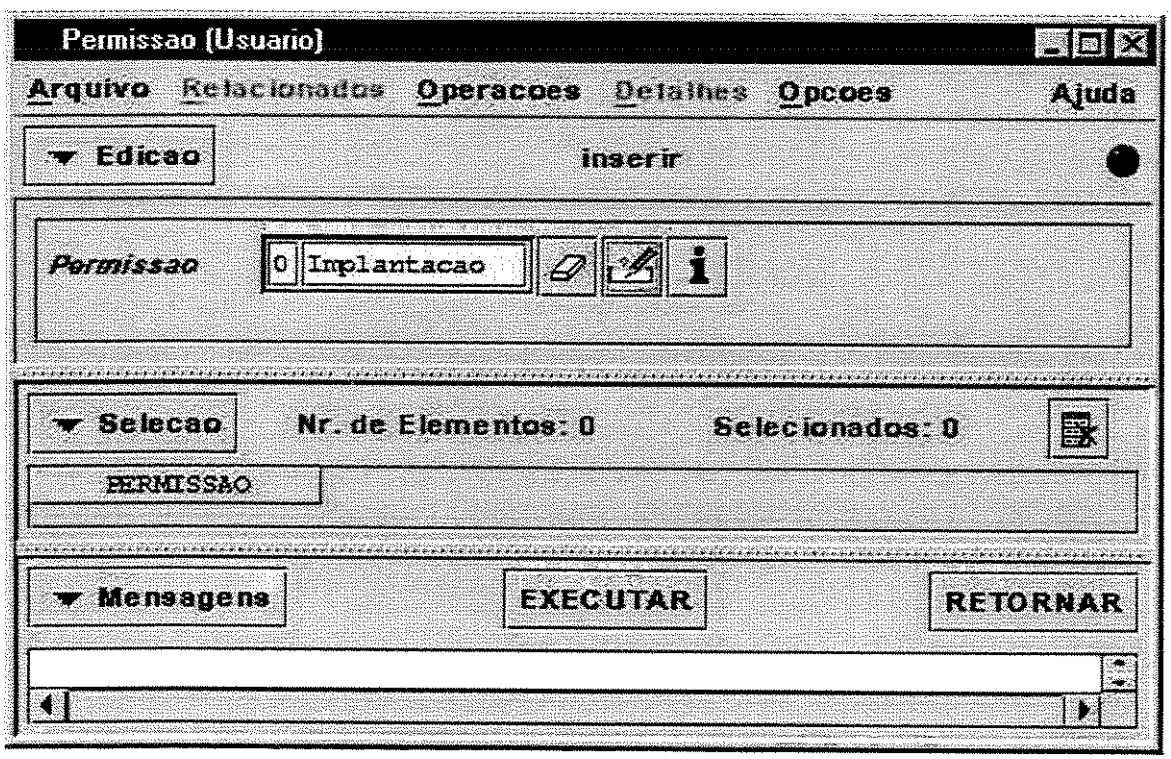


Figura 6.10: Tela de cadastramento de permissão.

6.3.3 Aspectos de implementação do controle de acesso

O controle de acesso do SAGRE foi implementado através de uma API para controle de acesso que responde a perguntas do tipo "O usuário com identificador X e senha Y tem acesso ao SAGRE?", "O usuário corrente tem permissão para executar a operação Z?". Esta API também tem a função de criptografar as senhas, recuperar o usuário corrente, recuperar a permissão do usuário, recuperar o servidor corrente e outras operações relacionadas ao controle de acesso.

O projeto desta API de controle de acesso levou em consideração algumas medidas importantes para aumentar a segurança do mecanismo:

- Criptografia da senha tão logo digitada.
- Funções de encapsulamento dos dados. Uma vez que a base de dados de trabalho tenha sido definida e a verificação de acesso ao módulo tenha sido bem sucedida, a identificação da base (para fins de controle de acesso) fica armazenada (encapsulada) no controlador de acesso e só pode ser modificada pela rotina de verificação de acesso ao módulo quando o usuário escolher outra base de trabalho. Deste modo, evita-se que uma aplicação mal-intencionada trapaceie e informe a base errada ao controlador de acesso. O mesmo argumento é aplicado ao ID do usuário e ao servidor corrente.
- O acesso de um usuário é verificado em duas etapas. Na primeira, o usuário é identificado no SAGRE pela verificação do *login* e senha. Fica a critério do programador da aplicação desabilitar todas as funcionalidades do módulo, exceto aquelas necessárias para abertura/seleção da base de dados. Na segunda, após a definição da base de trabalho, o acesso ao módulo é garantido pela verificação do direito de acesso ao módulo.

6.4 Mudando o SAGRE para controle de acesso por área geográfica

O controle de acesso para dados convencionais no SAGRE está implementado como descrito em 6.3. O próximo passo é o desenvolvimento do controle de acesso a partir de critérios geográficos. Para se ter um controle de acesso geográfico, é necessário que sejam realizadas em algum momento consultas geográficas, um procedimento muito caro em termos computacionais. Dessa forma, é preciso que se desenvolva um mecanismo de controle de acesso geográfico que mantenha o desempenho a níveis aceitáveis pelo cliente. Esta seção discute aspectos que devem ser incluídos no SAGRE para garantir este tipo de controle de acesso, usando o modelo proposto no capítulo 5.

6.4.1 Modificação no SAGRE/Adm

Atualmente o SAGRE/Adm permite inserir, remover e atualizar permissões do tipo $< s, -, m >$, onde s é o sujeito e m o perfil do usuário, ou seja, o conjunto de operações que o usuário pode executar sobre o banco. A primeira modificação a ser realizada é a alteração do SAGRE/Adm, com a conseqüente modificação de tabelas internas que armazenam perfis de usuários. Além desta modificação, o SAGRE/Adm deve ser responsável pelo gerenciamento de base de regras. Isto requer a criação de um novo módulo para inserção e remoção destas regras, conforme detalhadas nos algoritmos 5.1 e 5.2.

O SAGRE/Adm deve ser alterado para permitir inserir, atualizar e remover regras de autorização que indiquem o elemento espacial o a ser autorizado. Após a inserção do tipo de permissão (vide seção 6.3.1, deve ser possível inserir a região a qual o usuário com o tipo de permissão Y tem direito de acesso. Para isso deve ser criada uma tela onde o administrador poderá selecionar a região a ser autorizada, cadastrando no banco de dados o identificador desta região. A autenticação do usuário deve também sofrer alterações, já que passará por mais etapas de verificação. Uma possibilidade é incluir como parte do pedido de autenticação a recuperação, na base de regras, das regras $< s, x, m >$, para o usuário s. Isto poderá acelerar a verificação posterior de direito de acesso.

6.4.2 Controle de acesso geográfico de áreas de projeto (SAGRE/Cad)

Conforme mencionado, alterações na rede externa são efetuadas a partir de definição de "projetos" . A área de atuação de um projeto pode ser considerada como um polígono definido pelo usuário. A figura 6.11 apresenta uma cópia de tela com um projeto desenvolvido no SAGRE/Cad.

Atualmente, embora seja necessário criar um polígono para determinar os limites geográficos de um projeto, a delimitação desta área é apenas para fins de visualização, não impondo restrições sobre os objetos a serem alterados sob esse projeto. Assim, hoje o usuário pode fazer alterações em qualquer região visualizada. O controle de acesso geográfico permitirá que haja uma verificação para permitir alterações somente dentro da área delimitada pelo polígono definido para este projeto.

Isto significa que o módulo SAGRE/Cad deve ser modificado de forma a passar cada pedido de acesso pelo Gerenciador de Controle de Acesso proposto na dissertação. Este controle pode ser realizado de duas formas:

1. Uma única vez quando um projeto é criado. Esta criação é acompanhada de inserção de regras na base de regras. Neste caso, o acesso é garantido "para sempre" usando o algoritmo 5.3. Todos os usuários daquele projeto terão as permissões de acesso

Figura 6.11: Projeto desenvolvido no SAGRE/Cad.

do projeto. A vantagem é o tempo reduzido para verificação de direito de acesso. A desvantagem é a pouca flexibilidade se houver modificação no banco de dados ou na base de regras de autorização.

2. A cada pedido de acesso dentro de um projeto. Esta opção é mais flexível, porém mais cara.

A arquitetura e a solução propostas permitem implementar qualquer uma das duas opções.

Cada equipe de projeto passaria a ter um gerente de projetos com a tarefa de criar os projetos em sua área de atuação. Definida a área de projeto dentro dessa área de atuação, o gerente deve solicitar ao administrador criar via SAGRE/Adm novas permissões aos projetistas de sua equipe para que estes possam acessar o projeto criado. Os projetistas já devem estar autorizados a acessar a área de atuação que engloba a área deste projeto. Desta forma, a integração do modelo proposto com o SAGRE exigiria que toda nova criação de projeto passasse pela verificação de direito de acesso descrita no capítulo 5.

Além disso, o SAGRE/Cad deverá restringir a visualização do usuário somente a partes

do mapa às quais o usuário tiver permissão, ou seja, as suas áreas de atuação. Pode-se tentar utilizar o próprio cache gráfico para garantir todas as restrições de área.

6.4.3 Modificação no Gerenciador de Atributos

O Gerenciador de Atributos é um módulo que permite acessar e visualizar dados alfanuméricos. Ele se comunica diretamente com o SGBD Oracle, mas não com o Vision*, ou seja, não permite consultas geográficas. Quando se deseja consultar um determinado objeto no SAGRE/Cad, por exemplo estações telefônicas, o usuário pode selecionar a operação Filtrar e todos os objetos do tipo estações telefônicas são exibidos. Com o controle de acesso por área, deve-se exibir somente as estações localizadas nas áreas de atuação daquele usuário. Como as consultas são executadas pelo Gerenciador, isso requereria que o Gerenciador pudesse realizar consultas geográficas, o que não é possível hoje, já que não existe uma comunicação entre o Gerenciador e o SIG. Um dos empecilhos é a dificuldade de integração do Vision* com outras linguagens e o outro fator é o desempenho em se executar uma consulta geográfica neste momento. O controle de acesso por área geográfica irá também exigir modificar o Gerenciador de Atributos. Além de precisar ter acesso a consultas geográficas.

6.4.4 Controle de acesso por localidade no SAGRE/Tup

As localidades são porções menores que os municípios. Por exemplo, no caso do município de Campinas, temos as localidades de Sousas e Barão Geraldo.

A verificação de cobertura de uma localidade é gerenciada no SAGRE pelos chamados "consultores", que são responsáveis por estas áreas. Do ponto de vista lógico, tudo se passa como se um consultor coordenasse uma equipe que tenha acesso apenas à região geográfica daquela localidade (de novo, um polígono).

Os consultores devem visualizar e verificar a cobertura somente para as localidades sob sua responsabilidade. Devem ainda poder executar operações como inserir um TUP somente em suas áreas de atuação. O controle de acesso das localidades recai no mesmo problema de áreas específicas de projeto. A execução de operações também é similar aos problemas para atuação em áreas de projeto do SAGRE/Cad. Desta forma, a solução proposta para o SAGRE/Tup deve ser idêntica à do SAGRE/Cad.

6.5 Conclusão

Este capítulo mostrou o controle de acesso existente atualmente no sistema SAGRE e suas necessidades quanto ao controle de acesso geográfico, bem como as dificuldades para

implementá-lo. O capítulo aponta que módulos do SAGRE devem ser modificados para controle de acesso geográfico.

Capítulo 7

Conclusões e extensões

7.1 Contribuições

A dissertação apresentou um modelo de controle de acesso para bancos de dados geográficos, atendendo à crescente necessidade de prover maior segurança para dados geográficos. O mecanismo apresentado é genérico e pode ser adaptado segundo as necessidades de uma determinada área de aplicação. Um exemplo real foi a sua adaptação para as necessidades do sistema SAGRE.

Os modelos de autorização existentes foram apresentados e seus aspectos mais importantes foram detalhados. Através dos estudos realizados pôde-se concluir que as pesquisas sobre controle de acesso têm seguido três direções principais. A primeira diz respeito à política de controle de acesso seletivo, a segunda ao controle de acesso mandatório e a terceira trata do problema de controle de acesso para bancos de dados não convencionais.

Vale ressaltar ainda que inicialmente havia inúmeras pesquisas tentando estudar o controle de acesso seletivo. Entretanto, nos últimos anos pôde-se verificar um maior número de pesquisas sendo realizadas para o controle de acesso mandatório, devido às necessidades de um acesso hierarquizado, como nos órgãos militares e governamentais. Isso pode ser visto no número especial de 1996 da revista *IEEE Transactions on Knowledge & Data Engineering* sobre segurança em bancos de dados, onde cinco dos sete artigos relacionados ao assunto tratam de problemas relacionados ao controle de acesso mandatório [QL96, TS96, KPSN96, Mar96, DH96].

A proposta está baseada na definição de regras de autorização $< s, o, m >$. Os objetos o foram caracterizados como resultantes de uma consulta geográfica visando o controle de acesso. O mecanismo de controle de acesso proposto generalizou ao máximo os requisitos, sem se prender a nenhuma aplicação em especial. Finalmente, foi feita uma adaptação deste mecanismo genérico para atender aos requisitos do sistema SAGRE.

O controle de acesso para bancos de dados geográfico é uma abordagem ainda inédita.

As principais contribuições desta dissertação foram:

- estudo sobre os principais tipos de modelos de controle de acesso existentes;
- levantamento de requisitos para um controle de acesso para bancos de dados geográficos;
- definição de um modelo de autorização baseado em caracterização espacial;
- detalhamento dos aspectos de implementação deste modelo;
- proposta de adaptação e aplicação do mecanismo proposto para uma aplicação real, o Sistema SAGRE.

7.2 Extensões

Diversas extensões podem ser propostas a esta dissertação. Além da necessidade de sua implementação, existem vários aspectos teóricos a serem investigados. A seguir são enumerados alguns deles:

Controle de acesso seletivo: O modelo proposto não se baseia totalmente no controle de acesso seletivo, porque não tem uma administração de autorizações descentralizada. Entretanto, pode ser facilmente estendido para o controle de acesso seletivo, caso seja uma necessidade do tipo de aplicação. Como no modelo de controle de acesso seletivo a concessão e revogação de autorizações não é centralizada, seria necessário adicionar itens à tripla $< s, o, m >$, como por exemplo, quem concedeu a autorização e se o usuário tem a opção de repassar esta autorização, com possíveis valores, sim ou não.

Controle de acesso baseado em papéis: Em vários tipos de aplicações que utilizam dados geográficos, como no caso do sistema SAGRE, parece ser mais factível o uso da política de controle de acesso baseado em papéis, ou seja, as decisões de acesso são baseadas nas funções que um usuário pode executar dentro de uma organização. Os usuários não podem repassar autorizações a outros usuários de acordo com a sua vontade como no controle de acesso seletivo. A determinação do papel de um usuário e a alocação de tarefas para um papel é realizada por um administrador e isso é feito de acordo com regras de segurança da organização.

Controle de acesso com prioridades para sujeitos: Os casos de conflitos de regras, onde há dois ou mais sujeitos tentanto acessar o mesmo objeto, podem ser resolvidos classificando-se os sujeitos por níveis de prioridades. Isto pode determinar hierarquias de acesso. Neste caso, o conflito pode ser resolvido dando acesso ao sujeito de maior prioridade e negando ao outro. Isto permite usar uma hierarquia de acesso sem ter que utilizar o modelo de controle de acesso mandatório que exige bancos multiníveis.

Controle de acesso espaço-temporal: O modelo para dados geográficos pode ser estendido para atender também as necessidades temporais de seus dados, já que espaço e tempo estão geralmente relacionados. Por exemplo uma permissão envolvendo espaço-tempo seria "Ana pode acessar dados da cidade de Campinas das 8:00 às 17:00". Os algoritmos do capítulo 5 podem ser alterados para também verificar o intervalo temporal além da localização geográfica. O modelo espaço-temporal pode estender este modelo levando em consideração os estudos de Bertino e outros para mecanismos de controle de acesso temporal [BBFS96b, BBFS96a, BBFS97].

Controle de acesso por ações/operações: Algumas aplicações necessitam restringir o acesso por diversos tipos de operações e não somente "leitura" e "escrita" como proposto nesta dissertação. Essas operações podem ser tratadas como novos tipos de modo de acesso, o m da tupla $< s, o, m >$.

Inclusão de permissões caracterizadas por consultas geográficas *fuzzy:* O modelo não considerou operadores espaciais *fuzzy* devido à sua dificuldade de processamento. Entretanto, também pode ser estendido para permitir estes tipos de operadores, assim que houver ferramentas que possam processá-las.

Incorporação de permissões aninhadas O modelo pode facilmente ser estendido para incluir permissões complexas (vide seção 4.4). Uma permissão complexa pode ser entendida como $<< s1, o1, m1 >, < s2, o2, m2 >, m >$. Transcrevendo o exemplo da seção 4.4: "Todas as pessoas que têm acesso de leitura aos supermercados de Campinas têm igualmente acesso de escrita a todos os objetos acessáveis por supervisores".

A composição de permissões pode ser resolvida dentro do modelo proposto através da execução apropriada de consultas e identificação de objetos. Isto poderia ser incorporado ao algoritmo 5.4 de validação de pedidos de acesso. O algoritmo teria que ser alterado para executar também estas consultas complexas, talvez, dividindo-a em partes e executando no banco de dados para obter os sujeitos e objetos.

Consistência de conjuntos de autorizações: Não se tratou nesta dissertação de soluções para o problema de verificação de coerência entre permissões. Uma extensão seria estudar formas de resolver parcialmente este problema. A inclusão de uma permissão a um conjunto de permissões coerentes equivale ao seguinte:

Dado um conjunto de restrições coerentes, este conjunto continua coerente se mais uma restrição for adicionada?

Não é possível encontrar uma solução algorítmica definitiva, pois trata-se de um problema NP-completo.

Mecanismo de propagação de autorizações de controle de acesso para bancos de dados distribuídos: Do ponto de vista de controle de acesso, a grande diferença para bancos de dados distribuídos está na manutenção das autorizações. A execução das atualizações em diferentes ordens em diferentes nós (*sites*) pode gerar inconsistências, causando sérias falhas no controle de acesso.

Os bancos de dados distribuídos têm um certo número de nós, onde os dados são replicados, ou seja, cada nó mantém uma cópia de parte ou todo o banco de dados. A replicação de dados é feita com o objetivo de aumentar a disponibilidade dos dados e eficiência no acesso. Por essas mesmas razões é desejável replicar também as autorizações para controle de acesso, pois de nada adiantaria ter os dados localmente tendo que se validar as autorizações remotamente [PSJ94]. Por exemplo, suponha que Alice deseja acessar dados de uma região que está replicada localmente, mas a tabela de autorizações não está. Então, necessita-se de um acesso remoto para validar a autorização de Alice antes de autorizar o seu acesso aos dados. Obviamente isso não é eficiente e é necessário replicar também a tabela de autorizações (base de regras).

Como vimos no capítulo 2, há dois modelos de autorização principais utilizados para prover segurança para bancos de dados: controle de acesso seletivo e controle de acesso mandatório. Como o modelo mandatório tende a ser estático, ou seja, seus níveis de segurança praticamente não são alterados, acredita-se ser mais interessante no caso de bancos de dados distribuídos considerar apenas o modelo de controle de acesso seletivo, onde as autorizações tendem a ser dinâmicas.

A maneira tradicional de evitar inconsistências em um contexto distribuído é usar um protocolo de *commit* atômico, como o protocolo de *commit* de duas fases [PABG87]. Todavia, surgem vários problemas quando se tratam transações que acessam autorizações. Após um nó ter votado para fazer *commit*, ele está sujeito a um período de incertezas, durante o qual o nó não deve permitir o acesso a esses dados incertos. Uma falha de um nó ou uma falha de comunicação pode fazer com que uma transação seja abortada, o que é particularmente problemático para atualizações de autorizações. E finalmente, a recuperação após uma falha necessita de uma ação do usuário. Por exemplo, caso a

requisição de revogação de acesso feita pela usuária Alice seja abortada, a responsabilidade em refazê-la depende geralmente dela mesma.

Samarati, Ammann e Jajodia [PSJ94] propuseram um novo algoritmo otimista de controle de réplica especialmente para tratar a propagação dessas autorizações em bancos de dados distribuídos. Entretanto, faltam ainda detalhes de estratégias de comunicação entre os nós para trocar informações e saber quais requisições já foram processadas e quais ainda não foram.

Holliday e outros [JHA99] propuseram recentemente um novo algoritmo para gerenciar dados replicados fazendo um quorum epidêmico (*epidemic quorums*). Este novo algoritmo poderia ser adaptado para atender os requisitos da propagação de autorizações do controle de acesso seletivo e prencher justamente as lacunas que faltam no trabalho de Samarati e outros [PSJ94].

Referências Bibliográficas

[AJ96] P. Ammann and S. Jajodia. Ensuring atomicity of multilevel transactions. *IEEE Symposium on Security and Privacy*, 1996.

[BBFS96a] E. Bertino, C. Bettini, E. Ferrari, and P. Samarati. Supporting periodic authorizations and temporal reasoning in database access control. *Proc. 22nd Int. Conference on VLDB*, páginas 472–483, 1996.

[BBFS96b] E. Bertino, C. Bettini, E. Ferrari, and P. Samarati. A temporal access control mechanism for database systems. *IEEE Trans. Knowledge and Data Eng.*, 8(1):67–80, 1996.

[BBFS97] E. Bertino, C. Bettini, E. Ferrari, and P. Samarati. Decentralized administration for a temporal access control model. *Information Systems*, 22(4):223–248, 1997.

[BDPSN96] Ahmad Baraani-Dastjerdi, J. Pieprzyk, and Reihaneh Safavi-Naini. Security in databases: A survey study. February:1–39, 1996. *http://citeseer.nj.nec.com/baraani-dastjerdi96security.html.*

[Ber92] E. Bertino. Data hiding and security in an object-oriented database system. *Proc. IEEE Int. Conf on Data Eng.*, páginas 338–347, 1992.

[BHAE00] Elisa Bertino, Moustafa A. Hammad, Walid G. Aref, and Ahmed K. Elmagarmid. An access control model for video database systems. In *CIKM*, páginas 336–343, 2000.

[BP76] D. E. Bell and L. J. La Padula. Secure computer systems: Unified exposition and multics interpretation. Technical report, The Mitre Corp., 1976.

[CCH+96] G. Câmara, M. A. Casanova, A. S. Hemerly, G. C. Magalhães, and C. M. B. Medeiros. *Anatomia de Sistemas de Informação Geográfica.* 10^{a} Escola de Computação. 1996.

[DD79] Dorothy E. Denning and Peter J. Denning. Data security. *Computer Surveys*, 11(3):227–249, 1979.

[Den83] D. E. Denning. *Cryptography and Data Security*. Addison-Wesley Publising Company, 1983.

[DH96] H. S. Delugach and T. H. Hinke. Wizard: A database inference analysis and detection system. *IEEE Trans. Knowledge and Data Eng.*, 8(1):56–65, 1996.

[DSCP01] Ernesto Damiani, P. Samarati, S. De Capitani, and S. Paraboschi. Controlling Access to XML Documents. *IEEE Internet Computing*, December:18–28, 2001.

[EBFS94] E. Gudes E. B. Fernandez and H. Song. A model for evaluation and administration of security in object-oriented databases. *IEEE Trans. Knowledge and Data Eng.*, 6(2):275–292, 1994.

[EBM98] S. Jajodia E. Bertino and L. Mancini. Advanced transaction processing in multilevel secure file stores. *IEEE Trans. Knowledge and Data Eng.*, 10(1):120–135, 1998.

[EBS95] S. Jajodia E. Bertino and P. Samarati. Database security - research and practice. *Information Systems*, 20(7):537–556, 1995.

[EBS96] S. Jajodia E. Bertino and P. Samarati. Supporting multiple access control policies in database systems. *In Proc. IEEE Symp. On Security and Privacy*, páginas 94–107, 1996.

[EBS97] S. Jajodia E. Bertino and P. Samarati. An extended authorization model for relational databases. *IEEE Trans. Knowledge and Data Eng.*, 9(1):85–101, 1997.

[ECvO93] P. di Felice E. Clementini and P. van Oosterom. A small set of formal topological relationships suitable for end-user interaction. *Proceedings of the* 3^{rd} *Symposium Spatial Database Systems*, páginas 277–295, 1993.

[EDM95] S. Granado E. Dias and G. Magalhães. Uso de versões na garantia de consistência em ambientes mistos de projeto e operação. *Anais, X Simpósio Brasileiro de Banco de Dados*, páginas 321–334, 1995.

[EH90] M. Engenhofer and J. Herring. A mathematical framework for the definition of topological relationships. *Proceedings of the 4th International Symposium on Spatial Data Handling*, páginas 803–813, 1990.

[EN97] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems.* Addison Wesley, third edition, 1997.

[Fag78] R. Fagin. On authorization mechanism. *ACM TODS*, 3(3):310–319, 1978.

[Feu93] M. Feutchwanger. *Towards a Geographic Semantic Database Model.* Tese de Doutorado, Simon Fraser University, 1993.

[FK92] D. Ferraiolo and Richard Kuhn. Role-Based Access Control. *Proceedings of* 15^{th} *National Computer Security Conference*, 1992.

[Fre75] J. Freeman. The modelling of spatial relations. *Computer Graphics and Image Processing*, 4:156–171, 1975.

[Gut94] R. H. Guting. An Introduction to Spatial Database Systems. *The VLDB Journal*, 3(4):357–400, 1994.

[GW76] P. G. Griffiths and B. Wade. An authorization mechanism for a relational dabase system. *ACM TODS*, 1(3):243–255, 1976.

[IGC94] N. B. Idris, W. A. Gray, and R. F. Churchhouse. Providing dynamic security control in a federated database. *Proc.* 20^{th} *Int. Conference on VLDB*, páginas 13–23, 1994.

[Ihr00] Ken Ihrer. Database security. *Information Security*, September, 2000.

[JD94] D. Jonscher and K. R. Dittrich. An approach for building secure database federations. *Proc.* 20^{th} *Int. Conference on VLDB*, páginas 24–34, 1994.

[JHA99] D. Agrawal J. Holliday, R. Steinke and A. E. Abbadi. Epidemic quorums for managing replicated data. Technical report trcs 99-32, University of California at Santa Barbara, 1999. Department of Computer Science.

[KPSN96] S. Jajodia K. P. Smith, B. T. Blaustein and L. Notargiacomo. Correctness criteria for multilevel secure transactions. *IEEE Trans. Knowledge and Data Eng.*, 8(1):32–44, 1996.

[Mar96] D. G. Marks. Inference in MLS database systems. *IEEE Trans. Knowledge and Data Eng.*, 8(1):46–55, 1996.

[Mat00] S. P. Matias. Processamento de consultas ao banco de dados de biodiversidade do BIOTA. Tese de Mestrado, Universidade Estadual de Campinas, Dezembro 2000.

[NO] G. M. Nyanchama and S. L. Osborn. Mandatory security in an object oriented database. *Technical Report Number 317*, páginas 1–32.

[PABG87] V. Hadzilacos P. A. Bernstein and N. Goodman. *Concurrency control and recovery in database systems*, volume Reading of *MA*. Addison-Wesley, 1987.

[PSJ94] P. Ammann P. Samarati and S. Jajodia. Propagation of authorizations in distributed database systems. *Proceedings of the 2nd ACM Conference on Computer and communications security*, November(2(4)):136–147, 1994.

[QL96] X. Qian and T. F. Lunt. A mac policy framework for multilevel relational databases. *IEEE Trans. Knowledge and Data Eng.*, 8(1):3–15, 1996.

[RBKW91] F. Rabitti, E. Bertino, W. Kim, and D. Woelk. A model for authorization for next-generation database systems. *ACM TODS*, 16(1):89–131, 1991.

[SBJ96] P. Samarati, E. Bertino, and S. Jajodia. An authorization model for a distributed hypertext system. *IEEE Trans. Knowledge and Data Eng.*, 8(4):555–562, 1996.

[SSL99] S. Ravada A. Fetterer X. Liu S. Shekhar, S. Chawla and C. Lu. Spatial databases - accomplishments and research needs. *IEEE Trans. Knowledge and Data Eng.*, 11(1):45–55, 1999.

[ST90] P. D. Stachour and B. Thuraisingham. Design of LDV: a multilevel secure relational database management system. *IEEE Trans. Knowledge and Data Eng.*, 2(2):190–209, 1990.

[TS96] R. K. Thomas and R. S. Sandhu. A trusted subject architecture for multilevel secure object-oriented databases. *IEEE Trans. Knowledge and Data Eng.*, 8(1):16–30, 1996.

[WL82] P. F. Wilms and B. G. Lindsay. A database authorization mechanism supporting individual and group authorizations. *Distributed Data Sharing Systems*, páginas 273–292, 1982.