

INTEGRAÇÃO DO MÉTODO T.A.&.A E DO
AMBIENTE YPY NA ESPECIFICAÇÃO E
PROTOTIPAÇÃO DE UM SIE

por

Nina Edelweiss

José Palazzo M. de Oliveira

RP nº 127

Junho/90



UFRGS

SABi



05234385

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
Av. Osvaldo Aranha, 99
90.210 - Porto Alegre - RS - BRASIL
Telefone: (0512) 21-84-99/271999
Telex: (051) 2680 - CUF BR
FAX: (0512) 244164

Correspondência: UFRGS-CPGCC
Caixa Postal 1501
90001 - Porto Alegre - RS - BRASIL

UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

Editor: Taisy Silva Weber

Automação de Escritórios - SBU
Automação: Escritórios
Sistemas: Automação: Escritórios
Método T. A. & A.

ENPg 1 03.03.00-6

UFRGS CPD - PGCC BIBLIOTECA		
N.º CHAMADA: FL1783	N.º REG: 36503	DATA: / /
ORIGEM: D	DATA: 2 / 7 / 90	PREÇO: Cr\$ 1000,00
UNID: PGCC	FORM.: PGCC	

UFRGS

Reitor: Prof: TUISKON DICK

Pró-reitor de Pesquisa e Pós-Graduação: Prof. ABILIO A. BAETA NEVES

Coordenador do CPGCC: Prof. Ricardo A. da L. Reis

Comissão Coordenadora do CPGCC: Prof. Carlos Alberto Heuser

Prof. Clesio Saraiva dos Santos

Profª Ingrid J. Pôrto

Prof. José Mauro V. de Castilho

Prof. Ricardo A. da L. Reis

Prof. Sergio Bampi

Bibliotecária CPGCC/CPD: Margarida Buchmann

ÍNDICE

1. Introdução	1
2. Especificação de Sistemas de Informação de Escritórios	2
3. Método T.A.& A.	4
4. Ambiente YPY	6
5. Especificação semiformal de uma locadora de fitas de vídeo.	9
5.1 Identificação das tarefas	9
5.2 Identificação dos objetos	9
5.3 Detalhamento das tarefas e das atividades	10
6. Especificação formal e prototipação da locadora de fitas de vídeo	13
7. Conclusão	16
Referências Bibliográficas	18
Anexo 1	20

CONFIDENTIAL

RESUMO

O trabalho apresenta um exemplo de integração do método T.A.& A. [HOP88] com o ambiente YPY [EDE89a]. É apresentada a especificação e posterior prototipação de um Sistema de Informação de Escritório, uma locadora de fitas de vídeo. A análise do escritório é feita através do método T.A.& A., resultando na especificação semiformal do mesmo. Esta especificação é orientada ao projeto do banco de dados do sistema, através do modelo de dados implementado pelo ambiente YPY, utilizado na prototipação. A modelagem dos aspectos dinâmicos é feita através de condições de violação expressas como regras de lógica de primeira ordem.

ABSTRACT

The integration of the T.A.& A. method [HOP88] and the YPY environment [EDE89a] is here presented. The specification and prototyping of an Office Information System, a video rental shop, are developed. The T.A.& A. method is used in the office analysis, resulting in a semiformal specification. This specification is oriented towards the database project, using the data model implemented by the YPY environment, used in the prototyping phase. Dynamic aspects of the system are modeled through integrity violation conditions, presented as first-order logic rules.

1. INTRODUÇÃO

Um escritório é uma unidade funcional de processamento de informações [NEWS0]. Consiste em um número elevado de elementos (pessoas, documentos, etc) interagindo entre si de modos geralmente bem definidos. A comunicação entre as pessoas pode ser efetuada verbalmente, ou através da troca de mensagens e de documentos. O conhecimento é partilhado por todos os elementos do escritório. As atividades são desenvolvidas de modo concorrente e assíncrono. O processo de automação de um escritório consiste em fornecer ferramentas computacionais às pessoas que nele trabalham, de modo a auxiliá-las no trabalho rotineiro e nas atividades de decisão. O conjunto destas ferramentas suportando o trabalho dos usuários forma um Sistema de Informação de Escritório (SIE).

Este trabalho tem por finalidade apresentar a integração de um método de análise de SIEs denominado T.A.& A. (Tarefas, Atividades e Ações) [HOP88] com o ambiente YPY, de desenvolvimento de bancos de dados dedutivos temporais [EDE88,89b], ambos desenvolvidos no CPGCC/UFRGS (no desenvolvimento do primeiro participa também o CPGA/UFRGS). Esta integração é realizada através de um exemplo prático de especificação formal de um SIE - especificamente, de uma agência de locação de fitas de vídeo. Inicialmente é feita a análise do sistema a ser especificado, com o auxílio do método T.A.& A. Desta análise resulta uma especificação semiformal do sistema. A seguir é feita a especificação formal do sistema, tendo como base o modelo de dados implementado pelo ambiente YPY. Este ambiente é utilizado para construir um protótipo da especificação, com a finalidade de validá-la.

A Seção 2 apresenta alguns conceitos a respeito de especificações de SIE. Nas Seções 3 e 4 são apresentadas as principais características do método T.A.& A. e do ambiente YPY. Na Seção 5 é feita a especificação semiformal da agência de locação de fitas de vídeo. A Seção 6 mostra a especificação formal através

do ambiente YPY. As conclusões tiradas desta experiência de especificação e prototipação de um SIE são apresentadas na seção 7.

2. ESPECIFICAÇÃO DE SISTEMAS DE INFORMAÇÃO DE ESCRITÓRIOS

A implementação de um SIE inicia com a coleta de requisitos junto aos usuários. Da organização e análise destes requisitos resulta uma especificação informal ou semiformal do sistema, onde são identificadas as tarefas a serem executadas pelo sistema, os objetos manipulados e as transações possíveis sobre estes objetos.

Com base na especificação informal ou semiformal dos requisitos é construída a especificação formal do sistema. A especificação formal de SIE deve levar em conta a influência humana existente nestes sistemas, apresentando não só a estrutura das informações que serão manipuladas, mas também a semântica do ambiente em que são desenvolvidas as aplicações. Deve, ainda, representar as características de concorrência e de tempo, e ser independente da implementação, de modo a não sofrer influência de avanços tecnológicos [GR182]. O resultado desta especificação é um esquema conceitual do escritório, composto de um modelo conceitual, o qual descreve as suas entidades estáticas e dinâmicas, e de uma linguagem formal. As entidades estáticas descrevem os aspectos estruturais do escritório, tais como documentos, mensagens e agentes. As entidades dinâmicas descrevem a evolução das entidades estáticas no ambiente modelado, constituindo o fluxo de informações, definido pela sequência válida de atividades a serem executadas [HE188, PER88].

Para garantir a fidedignidade da especificação, o esquema conceitual deverá ser compreensível não só para o projetista do sistema, mas também para os seus usuários (projetistas de aplicações e usuários finais). Estes últimos é que validam o sistema, sugerindo modificações e identificando inconsistências. é

importante, portanto, que o método escolhido para a especificação não seja por demais complexo, o que dificultaria seu perfeito entendimento por parte de usuários não treinados. A validação das especificações também é efetuada por simulações da execução da especificação, através de processos de prototipação rápida. Os protótipos são também utilizados para definir interfaces entre o sistema e os usuários.

A última fase do desenvolvimento de um SIE é o projeto da arquitetura física, o qual também se baseia na especificação formal. Nesta fase são considerados os equipamentos existentes no mercado para possibilitar a efetiva implementação do sistema.

No exemplo aqui apresentado, inicia-se pela organização e análise de requisitos, considerando já efetuada a coleta dos mesmos. Esta análise é efetuada através do método T.A. & A. Dela resulta uma especificação semiformal do escritório analisado. A especificação formal é efetuada tendo como base o modelo de dados implementado pelo ambiente YPY, o qual também serve para a construção do protótipo.

As diversas visões do escritório ou dos diferentes aspectos de trabalho nele desenvolvido dão origem a vários modelos de especificação. Segundo [NEW80], estes podem ser classificados em modelos (1) de fluxo de informações (baseado em processos), que usam as atividades executadas concorrentemente no escritório como os elementos mais relevantes, ou seja, se concentram nos seus aspectos dinâmicos, como o GOFOR [RIC87] e o INSYDE [KIN85], (2) procedurais (baseados em agentes), nos quais o escritório é modelado sob ponto de vista das funções executadas pelos elementos ativos do ambiente (os agentes), (3) de tomada de decisão, que dão ênfase às atividades de tomada de decisão de gerentes e de outros empregados do escritório, (4) de bancos de dados, nos quais a ênfase principal recai nos dados armazenados e manipulados pelo sistema, e (5) comportamentais, os quais se fixam nos aspectos sociais do escritório, envolvendo situações e reuniões

que fazem parte das tarefas de processamento das informações do escritório. Segundo [BRA84, BAR85], os modelos mais comuns são aqueles baseados em bancos de dados, em processos e em agentes, além de modelos mixtos, que envolvem simultaneamente mais de uma visão do escritório. O modelo utilizado no exemplo aqui apresentado é baseado no banco de dados.

3. MÉTODO T.A. & A.

A organização e análise dos requisitos do sistema a ser especificado é efetuada através do método de análise T.A. & A. [HOP88]. Este método desdobra a análise de um escritório em três níveis hierárquicos básicos, as tarefas, as atividades e as ações, conforme mostrado na figura 1.

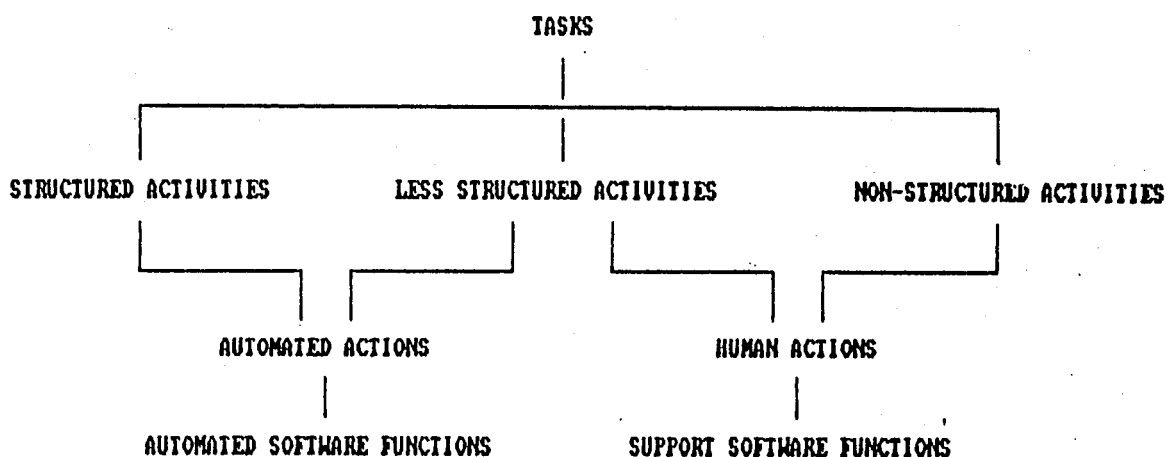


Fig. 1 - Composição hierárquica de tarefas, atividades e ações

As entidades do escritório são encaradas como objetos. São considerados objetos, por exemplo, documentos, fichas, mensagens, livros e textos.

As ações constituem os acessos elementares que podem ser efetuados sobre os objetos. São classificadas em ações automatizáveis ou não. As ações automatizáveis são executadas através de funções de software automáticas, representando acessos ao banco de dados sobre o qual o sistema modelado se baseia. As não automatizáveis são aquelas nas quais existe participação humana, sendo auxiliadas por funções de software de apoio.

As ações automatizáveis são primitivas do método, sendo aplicáveis a quaisquer objetos. Se classificam em dois tipos: (1) ações sobre classes de objetos, tais como criar uma classe, dar nome a uma classe, alterar atributos de uma classe; e (2) ações sobre instâncias, tais como criar uma instância, modificar valores de uma instância, remover uma instância, referenciar e obter valores da instância.

Uma atividade representa uma transação de usuário. É composta por : (1) um conjunto ordenado de ações; e (2) um conjunto de pré e de pós-condições para sua execução. As pré e pós-condições de cada atividade modelam a sequência de execução das atividades dentro da tarefa que está sendo considerada. Podem ocorrer casos de atividades concorrentes dentro de uma tarefa.

As atividades executadas nos SIE podem ser classificadas quanto ao grau de influência humana que apresentam, que corresponde diretamente à possibilidade de serem automatizadas. As atividades que são totalmente automatizáveis são chamadas de atividades estruturadas. Aquelas que possuem partes de trabalho que devem necessariamente ser desenvolvidas por pessoas, por envolverem ações não automatizáveis, como por exemplo, tomadas de decisão, são chamadas de atividades pouco estruturadas. Estas podem ser auxiliadas por computadores, através de funções de apoio como, por exemplo, indicação dos passos a serem seguidos, detecção e correção de erros. Os SIE apresentam, ainda, atividades que não podem ser executadas por computadores, devendo ser totalmente desenvolvidas por pessoas. São chamadas de atividades não estrutu-

radas. Estas também podem ser apoiadas por computadores, através de funções que possibilitem acessos ao banco de dados ou o emprego de ferramentas de simulação e de planejamento.

As tarefas são aquelas porções de trabalho que possuem uma significação própria dentro do escritório. Cada tarefa é composta por um conjunto de subtarefas ou de atividades. Podem, portanto, existir vários níveis de tarefas. No caso da modelagem de uma subtarefa, esta deve apresentar, além de sua decomposição em atividades ou outras subtarefas, um conjunto de pré e pós-condições para a sua execução.

A utilização do método T.A. & A. é efetuada através da seguinte sequência de passos: (1) identificação das tarefas que se quer especificar; (2) identificação dos objetos que serão manipulados; (3) detalhamento das tarefas em conjuntos de subtarefas ou de atividades, com a definição das pré e pós-condições de cada subtarefa e atividade; este passo deverá ser repetido até que todas as tarefas estejam desdobradas em atividades; e (4) detalhamento de cada atividade em um conjunto ordenado de ações.

4. AMBIENTE YPY

A especificação formal de um sistema pode ser efetuada definindo um banco de dados em termos do modelo de dados que implementa [LYN86]. As construções de modelagem estática do modelo, tais como os objetos, as classes de objetos e os relacionamentos, são utilizadas para definir o domínio semântico do sistema. As construções de modelagem dinâmica, tais como as operações do banco de dados, são utilizadas para definir a semântica das operações.

Na especificação do presente exemplo, uma locadora de fitas de vídeo, foi utilizado o modelo de dados implementado pelo Ambiente YPY [EDE88,89b]. Trata-se de um ambiente para desenvol-

vimento de bancos de dados dedutivos temporais, que pode ser utilizado em computadores PC-compatíveis. O seu modelo de dados define as propriedades estáticas do sistema através da definição de classes, de entidades definidas nestas classes, de relações binárias entre classes, de tuplas definidas nestas relações, e de um conjunto de condições de violação de integridade do banco de dados que limitam as possibilidades de definição de entidades e de tuplas. As propriedades dinâmicas são também modeladas através de condições de violação.

Constitui-se de um modelo de dados semelhante, em muitos aspectos, ao de TAXIS [MIL80], que é um modelo de dados orientado a entidades [NIX87]. Como naquele, as entidades podem ser criadas, destruídas e tem identidade única, isto é, são diferentes de quaisquer outras entidades passadas, presentes ou futuras. No modelo da ambiente utilizado isto também é válido para as classes, as relações e as tuplas.

Assim como no TAXIS, são admitidas somente relações binárias entre classes. Quando um objeto apresenta um número maior de propriedades (n), deverão ser formadas diversas relações binárias ($n+1$) para representá-las. Por exemplo, o objeto cliente apresenta as propriedades nome e endereço, entre outras. A modelagem destas propriedades será feita através de três relações, associadas através de uma nova propriedade comum a elas, por exemplo código: relação(código, cliente), relação(código, nome) e relação(código, endereço). Neste caso, cliente, nome, endereço e código são definidos como classes.

As condições de violação de integridade são avaliadas a cada ação efetuada, com a finalidade de verificar se houve violação de alguma regra de integridade e, em caso afirmativo, invalidar esta ação. Em comparação com métodos que modelam as propriedades dinâmicas através de redes, como por exemplo, através de Redes de Petri [RIC87], esta forma de modelagem representa uma situação complementar, isto é, enquanto nas redes são representa-

das as transições válidas entre estados, no presente caso são representados os estados inválidos alcançados através de transições inválidas.

As condições de violação são expressas através de regras de inferência, escritas em uma linguagem de lógica de predicados de primeira ordem. Nestas regras podem ser utilizados os dados representados segundo o modelo empregado, predicados predefinidos pelo Ambiente YPY, além de operadores PROLOG [STE86], linguagem de implementação do ambiente utilizado.

É muito importante, em SIE, a possibilidade de associar o tempo às informações, para representar, por exemplo, a duração de eventos, o calendário, as previsões, o tempo de vida de documentos ou de operações, e restrições temporais a determinadas atividades. A necessidade de inclusão do tempo associado às informações foi reforçada com o crescimento do uso de Sistemas de Suporte à Decisão. Nestes, é essencial a capacidade de manipulação do tempo e de informações históricas, principalmente para o planejamento empresarial, a investigação de relações causais e a análise retrospectiva [ARI86]. Como o ambiente utilizado implementa um banco de dados temporal, cada informação armazenada terá associada a si o momento de sua definição e, se for o caso, quando deixou de ser válida. Estas informações temporais poderão ser utilizadas na escrita das condições de violação.

A modelagem da concorrência também pode ser efetuado através do ambiente utilizado. Várias transições válidas e mutuamente exclusivas são permitidas no mesmo instante. As ações pertencentes a diferentes atividades poderão ser executadas intercaladamente. Através das condições de violação, entretanto, podem ser colocadas regras de ordem de execução para as transações.

5. ESPECIFICAÇÃO SEMIFORMAL DE UMA LOCADORA DE FITAS DE VÍDEO

A finalidade deste trabalho foi observar como se desenvolve a especificação e prototipação de um escritório real, utilizando para isto ferramentas desenvolvidas no CPGCC/UFRGS. O escritório escolhido é uma locadora de fitas de vídeo. Para não tornar o trabalho por demais extenso, serão modeladas somente algumas das muitas tarefas desenvolvidas neste tipo de escritório. A especificação semiformal, obtida através da utilização do método T.A. & A., é apresentada a seguir.

5.1 IDENTIFICAÇÃO DAS TAREFAS

As tarefas especificadas no presente trabalho são o cadastramento de clientes, o cadastramento de fitas e a locação de fitas a clientes desta locadora. Serão considerados somente os aspectos mais representativos de cada uma destas tarefas. Outras tarefas deste escritório seriam aquelas relacionadas com os funcionários da locadora, com publicidade e de controle financeiro.

5.2 IDENTIFICAÇÃO DOS OBJETOS

Os objetos manipulados na locadora são (1) fita, com as propriedades código, nome do filme e tipo do filme, (2) catálogo de fitas, formado por objetos fita, (3) cliente, com as propriedades código, nome e endereço, (4) bom_cliente, cadastro de bons clientes, formados por objetos cliente, (5) mau_cliente, cadastro de maus clientes, formado por nomes de clientes, e (6) locações, cadastro com informações a respeito das locações.

5.3 DETALHAMENTO DAS TAREFAS E DAS ATIVIDADES

A tarefa cadastramentar cliente é efetuada em dois cadastros. O primeiro armazena o nome e as informações a respeito de clientes que podem alocar fitas. O segundo guarda o nome de clientes indesejáveis, aos quais será vetada a entrada no primeiro cadastro. O controle destes dois cadastros é efetuado por duas subtarefas mutuamente exclusivas. A seguir será feita uma descrição informal de cada uma destas subtarefas, seguida do detalhamento em atividades e ações.

A subtarefa de cadastrar cliente novo é feita verificando, inicialmente, a pertinência do nome deste cliente no cadastro de clientes indesejáveis, o que bloquearia a sua inclusão no cadastro desejado; caso isto não ocorra, seu nome e informações a seu respeito (código, endereço, telefone, etc) são armazenados no cadastro correspondente.

Atividades:

a1. Definir código para o cliente

pré-condição: foi feito o pedido de registro de um cliente;

ação: obter código;

pós-condição: este código não aparece como código de cliente em nenhum dos dois cadastros de clientes.

a2. Registrar cliente no cadastro bom_cliente

pré-condição: o nome e o código deste cliente ainda não constam do cadastro bom_cliente;

ação: criar instância de bom_cliente, com código, nome e endereço;

pós-condição: o cadastro bom_cliente apresenta uma instância com os dados deste cliente.

A subtarefa de cadastrar cliente indesejável é executada verificando, inicialmente, a existência do nome deste cliente no cadastro de bons clientes da locadora; caso esta ocorrer, todas as informações deste cliente deverão ser retiradas deste cadastro. A seguir é efetuada a inclusão do nome no cadastro desejado.

Atividades:

a3. Retirar cliente do cadastro bom_cliente

pré-condição: este nome consta do cadastro bom_cliente;

ação: remover instância correspondente a este nome do cadastro bom_cliente;

pós-condição: o nome não consta do cadastro bom_cliente.

a4. Incluir nome no cadastro mau_cliente

pré-condição: este nome não consta de nenhum dos dois cadastros de clientes;

ação: criar instância do cadastro mau_cliente, com o nome.

A tarefa de cadastramentrar fita consiste em incluir os dados referentes a uma fita disponível para locação no cadastro de fitas. É armazenado o nome do filme e o tipo em que pode ser classificado (aventura, terror, infantil, etc.). Cada fita terá um código próprio.

Atividades:

a5. Definir código da fita

pré-condição: existe uma fita nova para ser cadastrada;

ação: obter código para fita;

pós-condição: o valor obtido não aparece como código de nenhuma fita no catálogo de fitas.

a6. Incluir fita no catálogo de fitas

pré-condição: nenhuma fita deste catálogo apresenta estes mesmos dados;

ação: criar instância de catálogo com código da fita, nome do filme e tipo do filme.

pós-condição: existe uma instância deste catálogo com estes dados.

Na tarefa de locar fitas devem ser considerados diversos fatores, de acordo com a semântica das informações armazenadas. É permitida, a um mesmo cliente, a locação de mais de uma fita, inclusive em datas diferentes. Um pedido de locação de uma fita por uma determinada pessoa somente será atendido se as seguintes pré-condições forem atendidas:

- c1. a pessoa for um cliente cadastrado no escritório;
- c2. a pessoa não estiver de posse de alguma fita por um período superior a 30 dias;
- c3. a fita estiver cadastrada (existir esta fita nesta locadora);
- c4. a fita estiver disponível para aluguel.

As pré-condições c1 e c3 poderão ser verificadas através de pesquisas nos cadastros correspondentes. A pré-condição c2 poderá ser verificada através de uma pesquisa das fitas locadas por este cliente, e a pré-condição c4, através das informações relativas às fitas locadas.

Na efetivação de locação de uma fita deverão ser guardados o código da fita e o código do cliente, o dia em que foi iniciada a locação, além de informações adicionais a respeito desta atividade, não utilizadas no atual exemplo.

Atividade:

a7. Registrar locação

pré-condição: foi feito pedido de locação com dados de código da fita, código do cliente e data de início da loca-

ção e nome do cliente pertence ao cadastro de bom_cliente e não existe locação deste cliente com início a mais de 30 dias e código da fita aparece no catálogo de fitas e código da fita não aparece em nenhuma das locações atuais;
ação: criar instância do cadastro de locações;
pós-condição: foi feito o registro da locação.

A tarefa de devolver fita será executada através da remoção, do cadastro de locações, das informações relativas a esta locação. O controle financeiro da locação, se realizado, seria representado por ações complementares nesta mesma atividade.

Atividade:

a8. Desfazer locação

pré-condição: existência de uma locação com este código de fita e existência de data de retorno;

ação: remover a instância correspondente a esta locação do cadastro correspondente;

pós-condição: esta locação não está mais registrada.

6. ESPECIFICAÇÃO FORMAL E PROTOTIPAÇÃO DA LOCADORA DE FITAS DE VÍDEO

As tarefas, atividades e ações, definidas informalmente, são agora formalizadas através do modelo de dados do ambiente YPY.

Neste ambiente, as ações sobre tipos são: (1) a criação de uma classe com a definição simultânea do tipo de seus elementos (inteiros, reais ou simbólicos); (2) a definição de uma classe derivada através de operações entre classes já definidas (união, interseccção ou complemento); (3) a remoção de uma classe, o que implica na automática remoção de todas as entidades nela definidas, de todas as relações e classes derivadas que a en-

volvem e de todas as tuplas definidas nestas relações; (4) a definição de uma relação entre duas classes, com a simultânea definição do tipo de relação (total ou parcial); (5) a definição de uma relação derivada através de operações entre relações já definidas (união, intersecção, complemento, composição e inversa); (6) a remoção de uma relação com a automática remoção de todas as suas tuplas e relações derivadas; e (7) a referência a classes e relações.

As ações sobre instâncias são: (1) a definição de entidades em uma classe; (2) a remoção de entidades de uma classe; (3) a definição de tuplas em uma relação; (4) a remoção de tuplas de uma relação; e (5) o acesso a entidades e a tuplas.

A modelagem das informações no modelo de dados lógico do ambiente utilizado requer que sejam criadas classes (simbólicas ou numéricas) para armazenar as entidades válidas e relações binárias entre entidades destas classes. As classes, com o seu tipo associado, indicado entre parêntesis, são as seguintes:

- nome_cliente (simbólica) : nomes dos clientes do cadastro de bons clientes - aqueles que podem locar fitas;
- mau_cliente (simbólica) : nomes dos clientes do cadastro de maus clientes - aqueles que não são aceitos como clientes do escritório;
- cod_cliente (numérica inteira) : códigos dos clientes;
- end_rua (simbólica) : nomes de ruas;
- end_nr (numérica inteira) : número de endereços.
- nome_filme (simbólica) : nome de filme;
- tipo_filme (simbólica) : tipo do filme;
- cod_fita (numérica inteira) : código de uma fita.

As relações definidas e as duas classes relacionadas por cada uma delas são as seguintes:

- cliente (cod_cliente , nome_cliente);
- rua (cod_cliente , end_ rua);
- numero (cod_cliente , end_nr);
- apto (cod_cliente , end_nr).
- fita (cod_fita , nome_filme);
- tipo (cod_fita , tipo_filme).
- alugado (cod_fita , cod_cliente);

Não foi necessário criar representações explícitas para a data de locação pois a informação temporal é acrescentada a cada informação pelo próprio sistema utilizado.

Como este ambiente realiza o controle de integridade através de condições de violação, condições estas avaliadas após a execução de uma atividade, é necessário representar os estados alcançados pelo sistema em todos os possíveis casos de não atendimento das pré-condições definidas. Cada um destes estados é representado por uma condição de violação que, em caso de ser satisfeita, envia uma mensagem ao usuário indicando qual a atividade incorreta executada. As pós-condições também são representadas por condições de violação.

Por exemplo, estados inválidos do sistema, relativos ao cadastramento de clientes, são alcançados quando (1) for incluído o nome de um novo cliente na classe nome_cliente, estando este nome cliente já definido na classe mau_cliente; (2) for incluído o nome de um cliente indesejável na classe mau_cliente, estando este nome na classe nome_cliente; ou (3) um mesmo nome for incluído duas vezes em uma mesma classe. Esta última condição nunca será alcançada pois o sistema não permite duas entidades com nome igual em uma mesma classe. No Anexo 1 são apresentadas as atividades que levam a violações do sistema do exemplo apresentado.

A partir destas atividades inválidas são construídas as condições de violação. Por exemplo, a condição abaixo testa a execução das duas atividades acima descritas:

```
viola :- entidade(nome_cliente,X),
        definida(nome_cliente,X),
        entidade(mau_cliente,X),
        definida(mau_cliente,X),
        mensagem('NOME JA CONSTA DE OUTRO CADASTRO.').
```

Já a condição correspondente à condição c2, que testa se o cliente possui uma fita há mais de 30 dias, onde o tempo está explícito, seria representada por:

```
viola :- tupla(alugado,Fita1,Cliente),
        existe(alugado,Fita1,Cliente,Tempo1),
        tupla(alugado,Fita2,Cliente),
        existe(alugado,Fita2,Cliente,Tempo2),
        antes(Tempo2,Tempo1),
        intervalo(Tempo2,Tempo1,Dias),
        Dias>30,
        mensagem('ESTE CLIENTE ESTA COM FITA EM ATRASO.').
```

Como as condições são avaliadas após a execução de uma atividade, no caso da locação de uma fita, o fato de um cliente estar de posse de uma fita em atraso será detectado quando ele possuir duas fitas (a atual - Fita1, e a atrasada - Fita2), sendo o intervalo entre o dia atual e o de locação da fita anterior superior a 30 dias. Os predicados definida, mensagem, existe, antes e intervalo são pré-definidos no ambiente [EDE89b].

7. CONCLUSÃO

A especificação de SIE é uma das atividades mais importantes do seu desenvolvimento, sendo a base das fases de prototipação e de seu projeto final. A escolha de um formalismo adequado é de suma importância. Além da capacidade de representar o sistema com exatidão, tanto em suas características estáticas como nas dinâmicas, este formalismo deverá apresentar facilidades para a

posterior validação da especificação.

Neste trabalho foi testada a utilização de dois métodos desenvolvidos no CPGCC/UFRGS na especificação de um escritório real. O método T.A. & A. foi utilizado para a análise e consequente especificação semiformal do sistema, e o ambiente YPY para a sua especificação formal e posterior prototipação. Ambos os métodos se mostraram adequados para alcançar os objetivos propostos. O ambiente YPY está totalmente implementado, permitindo a criação de protótipos de especificações. A formalização da especificação através de seu modelo de dados é uma consequência natural da especificação semiformal obtida a partir do método T.A. & A.

Este trabalho é parte integrante de um projeto de visa a fornecer um ambiente completo para a especificação e a implementação de SIE. Este ambiente deverá apresentar ferramentas que apoiem, de forma integrada, todas as fases de desenvolvimento de um SIE, desde a coleta de requisitos até sua implementação física.

REFERÊNCIAS BIBLIOGRÁFICAS

- [ARI86] ARIAV,G. A Temporally oriented data model. ACM Transactions on Database Systems, 4(4):499-527, Dec. 1986.
- [BAR85] BARBIC,F.; CERI,S.; BRACCHI,G. Modeling and integrating procedures in office information systems design. Information Systems, New York, 10(2):149-68, Apr. 1985.
- [BRA84] BRACCHI,G. & PERNICI,B. The Requirements of office systems. ACM Transactions on Office Information Systems, 2(2):151-170, Apr. 1984.
- [EDE88] EDELWEISS,N. & COSTA,A.C. da R. Um Ambiente para desenvolvimento de protótipos de bancos de dados dedutivos. Porto Alegre, CPGCC/UFRGS, 1988. (RP nº 97).
- [EDE89a] EDELWEISS,N. & ROCHA,A.C. da R. Ambiente para desenvolvimento de bancos de dados dedutivos temporais. In: SIMPÓSIO BRASILEIRO DE BANCOS DE DADOS, 4., Campinas, abr. 1989. Anais. Campinas, R. Vieira Gráfica e Ed., 1989. p.163-173.
- [EDE89b] EDELWEISS,N. Introdução do tempo no ambiente para desenvolvimento de protótipos de bancos de dados dedutivos. Porto Alegre, CPGCC/UFRGS, 1989. (RP nº 114).
- [GRI82] GRIETHUYSEN,J.J. et al. Concepts and terminology for the conceptual schema. ISO/TC97/SC5/WG3 Report, 1982.
- [HEI88] HEIJMINK,F. et al. Development of tools for designing OIS. In: BULLIGER,U.-J. et al., ed. Information technology for organizational systems. Elsevier Science Publishers B.V., Brussels, 1988. p.66-73.
- [HOP88] HOPPEN,N.; OLIVEIRA,J.P.M.; LIMA,L.M.R. Metodologia para modelagem de escritório baseada no nível de estruturação das atividades. In: REUNIO ANUAL DA ASSOCIAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO EM ADMINISTRAÇÃO, 12., Natal, 26-28 set. 1988. Anais. Natal, 1988. p.247-263.
- [KIN85] KING,R. & McLEOD,D. A Database design methodology and tool for information systems. ACM Transactions on Of-

- Office Information Systems, 3(1):2-21, Jan. 1985.
- [LYN86] LYNGBAEK,P. & KENT,W. A Data modeling methodology for the design and implementation of information systems. In: INTERNATIONAL WORKSHOP ON OBJECT-ORIENTED DATABASE SYSTEMS, Pacific Grove, California, Sept. 23-26, 1986. Proceedings. New York, IEEE, 1986. p.6-17.
- [MIL80] MYLOPOULOS,J.; BERNSTEIN,P.A.; WONG,H.K.T. A Language facility for designing database-intensive applications. ACM Transactions on Database Systems, 5(2):185-207, June 1980.
- [NEW80] NEWMAN,W. Office models and office systems design. In: INTEGRATED Office Systems - Burotics. Amsterdam, North-Holland, 1980. p.3-10.
- [NIX87] NIXON,B. et al. Implementation of a compiler for a semantic data model: experiences with TAXIS. In: ACM ANNUAL CONFERENCE ON MANAGEMENT OF DATA, San Francisco, May 27-29, 1987. SIGMOD RECORD, 16(3):118-31, Dec. 1987.
- [PER88] PERNICI,B. et al. C-TODOS : an automatic tool for system conceptual design. Milan, Politecnico di Milano, Italy. Oct. 1988.
- [RIC87] RICHTER,G. et al. Generic office frame of reference GOFOR. St. Augustin, GMD, F.R.Germany, 1987. (Esprit Project 56: Functional Analysis of Office Requirements).
- [STE86] STERLING,L & SHAPIRO,E. The Art of Prolog - advanced programming techniques. Harvard, The MIT Press, 1986.

ANEXO 1 - AÇÕES DE VIOLAÇÃO

- a) incluir o mesmo nome de cliente duas vezes na classe cliente;
- b) incluir o mesmo nome de cliente duas vezes na classe mau_cliente;
- c) incluir o mesmo nome de rua duas vezes na classe end_rua;
- d) incluir o mesmo nome de filme duas vezes na classe nome_filme;
- e) incluir o mesmo tipo de filme duas vezes na classe tipo_filme;
- f) incluir o nome de um novo cliente na classe nome_cliente estando o nome deste cliente definido na classe mau_cliente;
- g) incluir o nome de um cliente indesejável na classe mau_cliente estando este nome na classe nome_cliente;
- h) incluir uma tupla na relação cliente associando um nome de um cliente a um código já existente em outra tupla desta mesma relação;
- i) incluir uma tupla na relação cliente, associando um código a um nome já existente em outra tupla desta mesma relação;
- j) incluir uma tupla na relação rua, associando uma rua a um código que não aparece em nenhuma tupla da relação cliente;
- k) incluir uma tupla na relação numero, associando um número a um código que não aparece em nenhuma tupla da relação cliente;
- l) incluir uma tupla na relação apto, associando o número de um apartamento a um código que não aparece em nenhuma tupla da relação cliente;
- m) incluir uma tupla na relação bairro, associando o nome de um bairro a um código que não aparece em nenhuma tupla da relação cliente;
- n) incluir uma tupla na relação filme, associando o nome de um filme a um código de fita já existente em outra tupla desta mesma relação;
- o) incluir uma tupla na relação tipo, associando o tipo de um filme a um código de fita já existente em outra tupla desta mesma relação;
- p) incluir uma tupla na relação tipo, associando o tipo de um filme a um código de fita que não aparece em nenhuma outra tupla da relação filme;

- q) incluir uma tupla na relação alugado, associando o nome de um cliente ao código de uma fita que já está alugada, isto é, cujo código já aparece definido em alguma outra tupla desta relação;
- r) incluir uma tupla na relação alugado quando existe alguma outra tupla em alugado com o código do mesmo cliente, tendo sido efetuado a locação em uma data anterior a 30 dias da atual;
- t) remover uma entidade da classe nome_cliente, quando existir alguma locação em seu nome, ou seja, quando existir alguma tupla da relação alugado na qual aparece o código associado a este nome em alguma tupla de cliente;
- u) remover uma entidade da classe nome_cliente quando existir alguma tupla nas relações rua, numero ou apto associada ao código relacionado a este nome através da relação cliente.
- v) remover alguma rua da classe end_rua, se esta estiver em alguma tupla da relação rua;
- x) remover o nome de um filme da classe nome_filme, se existir locação de alguma fita com este filme, isto é, se existir alguma tupla em alugado com o código de alguma fita que está associado a este nome de filme através de uma tupla da relação fita;
- y) remover alguma entidade da classe tipo_filme, se este tipo estiver associado a algum filme através de uma tupla da relação tipo.