

PROJETO DO ESQUEMA CONCEITUAL PARA SISTEMAS DE INFORMAÇÃO: UMA ABORDAGEM USANDO A METODOLOGIA THM.

C.L.G. MEDEIROS e U. SCHIEL

SUMÁRIO

É apresentada a primeira das duas fases de uma metodologia para o projeto de esquemas conceituais para sistemas de informação(SI). Esta metodologia modela tanto os aspectos estáticos como os dinâmicos do SI, com ênfase especial nos últimos. Na primeira fase é desenvolvido um Modelo de Requisitos para descrever o SI através de Grafos Estruturais e de Comportamento baseados em modelo (restrito) semântico de dados.

A metodologia está sendo aplicada com sucesso no desenvolvimento de um SI para uma pequena biblioteca. Como ilustração, é apresentado o Modelo de Requisitos desse sistema.

ABSTRACT

The first of two phases of a methodology for designing conceptual Schemes for Information Systems(IS) is presented. This methodology models both the static and the dynamic aspects of IS, with special emphasis on the latter. The first phase develops a Requirements Model which describes the IS by using Behaviour and Structural Graphs, both based on a restricted semantic data model. The methodology is successfully being applied in the development of an IS for a small library. As an illustration, the Requirements Model for such system is developed and presented.

Camilo de Lelis Gondim Medeiros,

Tecnólogo em Processamento de Dados-UFPb/Fev/77, atualmente elaborando Dissertação de Mestrado em Sistemas e Computação e Professor Assistente do DSC/UFPb. Áreas de Interesse: Banco de Dados, Sistemas de Informação, Linguagens de Programação, Compiladores. UFPb/DSC/Av. Aprígio Veloso, 882/Bodocongó, 58.100/Campina Grande/Pb. Tel. (083)7222 - Ramal: 510.

Ulrich Schiel,

Bacharel em Matemática-Mackenzie/SP.(1971), Aperfeiçoamento em Matemática/Impa/RJ.(1974). Mestrado em Informática/PUC/RJ.(1977), Doutorado em Informática/Universidade de Stuttgart Alemanha(1984), Professor do DSC/UFPb. Áreas de interesse: Banco de Dados, Sistemas de Informação. UFPb/DSC/Av. Aprígio Veloso, 882/Bodocongó, 58.100/Campina Grande/Pb.

Tel. (083)7222 - Ramal: 510.

1. INTRODUÇÃO

Durante os últimos anos foram desenvolvidas várias abordagens para modelagem de sistemas de Informação (SI) (compare com [9]). Todas elas focalizam o problema de derivar um esquema conceitual representando o universo de discurso que pode ser manuseado pelo SI. Para derivar tal esquema conceitual é requerida uma metodologia de projeto para dar suporte tanto ao especialista de aplicação quanto ao Administrador de Banco de Dados (ABD) durante o processo de mapeamento. Ao desenvolver uma metodologia de projeto, deve ser encontrado um equilíbrio entre as necessidades do especialista de aplicação e o ABD. Para dar suporte ao especialista de aplicação durante o processo de estabelecer os requisitos para o SI a metodologia tem de oferecer conceitos orientados mais para a aplicação do que para o banco de dados. Uma vez que o especialista de aplicação é um expert na área de aplicação que deve ser coberta pelo SI, e não um expert em bancos de dados. Por outro lado, o ABD está interessado primariamente nos conceitos de modelagem orientados para banco de dados para descrever os requisitos de SI a ser desenvolvido, para ter uma base bem estabelecida para derivar o esquema conceitual. Para evitar misturas entre o especialista de aplicação e o ABD, é muito importante prover conceitos de modelagem na metodologia de projeto que possam ser usados como uma base de comunicação entre o especialista de aplicação e o ABD.

Para solucionar este conflito Horndasch et.al.([7]) propuseram uma metodologia de projeto que será chamada de metodologia THM. Esta metodologia será apresentada no presente trabalho com algumas pequenas melhorias e ilustramos sua utilização para a modelagem de um sistema de controle do acervo bibliográfico de uma biblioteca setorial. Por limitações de espaço o sistema não será descrito até os últimos detalhes, mas sim, com detalhes suficientes para ilustrar a metodologia.

A metodologia THM é dividida em duas fases (veja a figura 1):

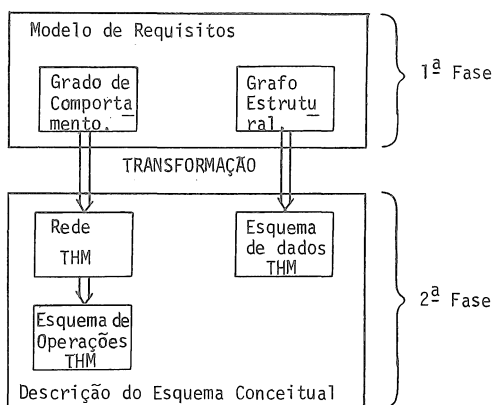


Figura 1: Visão da abordagem em duas fases.

(1) A primeira fase é orientada para a análise dos requisitos para o SI e estabelecimento de um modelo inicial para o SI. Uma vez que os aspectos dinâmicos de concorrência são muito decisivos em um SI, por exemplo quando ele é usado em um ambiente de escritório (veja [2]), é dada ênfase a estes aspectos dinâmicos durante a primeira fase do projeto. Entretanto, desde que o objetivo máximo da metodologia é chegar a uma descrição do esquema conceitual baseada em conceitos do modelo semântico de dados, o processo de análise efetuado pelo especialista de aplicações deve ser guiado também por estes conceitos dos modelos semânticos de dados. Usando diversos passos de refinamento a primeira fase resulta no que é chamado de Modelo de Requisitos (MR) que é composto de duas partes: um Grafo de Comportamento(GC), especificando os eventos induzidos pelo comportamento funcional do SI e o fluxo de informações entre funções ao usar uma rede canal/agência generalizada (compare com [12]) e um Grafo Estrutural(GE) representando a estrutura das informações manipuladas pelo SI. O GE é construído usando um modelo semântico de dados restrito a relações binárias.

(2) Durante a segunda fase o MR é transformado em uma rede de alto nível (chamada Rede THM) provendo um modelo conceitual totalmente formal do SI. Na rede THM o modelo semântico de dados THM (Modelo Temporal - Hierárquico) ([14], [15]) é usado para descrever os dados que são consumidos/produzidos pelas transições na rede THM. Entre os vários conceitos de Petri-nets foram selecionados conceitos de redes de predicado/transição (PrT-net) ([13]) e modificados apropriadamente para integrar os conceitos de Petri-nets com os conceitos de THM. Em adição à rede THM, é provida uma descrição estática do esquema conceitual descrevendo a estrutura dos dados manipulados pela rede THM.

Têm sido desenvolvidas várias metodologias para a modelagem e projeto dos sistemas de informação. Em [8] é proposta a metodologia ISAC. Entretanto, os grafos de estrutura usados na ISAC não fornecem meios para prover uma especificação detalhada das condições de ativação nem cobrem aspectos de concorrência.

A abordagem descrita em [5] é totalmente descritiva e não inclui quaisquer aspectos operacionais. A metodologia THM tem algumas similaridades com a abordagem introduzida em [10] e [12] que usa as chamadas IML - PrT-nets inscritas. Entretanto, ela difere desta abordagem em diversos aspectos importantes:

- (i) Em oposição ao MR de THM a rede canal/agência, usada como um primeiro modelo na abordagem IML, não inclui quaisquer conceitos de modelagem de dados e simplesmente oferece um canal como uma unidade funcional que armazena informação.
- (ii) A inscrição IML de PrT-nets é baseada nos Conceitos de Gerenciamento de Informação (CGI) [11] que não provêem a poderosa modelagem (p. ex. os princípios de abstração) dos modernos modelos semânticos de dados como p. ex. no modelo de dados THM, que é usado na metodologia.
- (iii) Desde que os aspectos de dados da metodologia THM são modelados em ambas as fases do projeto, é obtido um desenvolvimento mais paralelo de dados e atividades.

(iv) Em [12] faltam orientações precisas como transformar o primeiro modelo, o modelo canal/agência, no segundo modelo, o IML - PrT-net inscrito. Na metodologia THM são da das regras para uma transformação sistemática do MR na rede THM.

As partes subsequentes do presente trabalho são organizadas como segue. Na seção 2 é apre sentada uma breve introdução ao Modelo de Dados Temporal-Hierárquico (THM). Na seção 3 é introduzido o Modelo de Requisitos consistindo do Grafo de Comportamento e do Grafo Estru tural. Adicionalmente, descreve-se os diferentes passos de desenvolvimento que são usados para gerar o MR. Para propósitos de ilustração, é usado o problema de automação das ativi dades de uma pequena biblioteca. Em um artigo futuro, proveremos a definição da nossa re de de alto nível, i.e., a rede THM para o sistema citado e discutiremos as regras para transformação do MR em rede THM e a descrição da estrutura de dados THM.

2. O MODELO DE DADOS TEMPORAL-HIERÁRQUICO (THM)

Desde que a metodologia de projeto apresentada resulta em esquemas baseados no modelo se mântico de dados THM, é dada uma breve descrição dos conceitos mais importantes de THM. Uma descrição detalhada e também uma formalização podem ser encontradas em [14] e [15]. Em THM tanto os conceitos de modelagem bem-conhecidos de diversos modelos semânticos de da dos, p. ex. SDM ([6]) e SHM + ([1]), quanto novos conceitos como p. ex. conceitos de mode lagem de tempo, são integrados em uma abordagem consistente.

A parte estática de THM, o esquema de dados, é baseada nos conceitos "entidade", "classe (de entidade)", "relação (binária)". São distinguidos três tipos de relação: (a) relações membro-a-membro descrevendo relações entre membros de duas classes, (b) relações classe-membro especificando relações entre uma classe como um todo e membros de outra classe, e (c) relações de classes descrevendo relações entre duas classes. Para ser capaz de estabe lecer uma hierarquia de classes de entidades são incluídos os três princípios de abstra ção: generalização, agregação e agrupamento em THM. Uma vez que a ênfase é dada aos aspec tos de modelagem de tempo, THM não só provê conceitos para descrever o estado atual de en tidades e relações, mas também, oferece meios de especificar o tempo de vida de entidades e de relacionamentos, armazenando seu passado e indicando o seu futuro.

A parte dinâmica de THM, o esquema de operações, oferece conceitos para descrever opera ções que podem ser usadas para transformar um estado do banco de dados em um estado suces sor. Se o sucesso da execução de uma operação depende de requisitos que devem ser atendi dos pelo estado atual da base de dados e/ou seu estado sucessor, são incluídas pré-condi ções e pós-condições respectivamente, na definição de uma operação. Uma operação é cons truída de operações primitivas (inserir, deletar, para entidades; estabelecer, remover, pa relações) e operações complexas já definidas. Para descrever modificações no estado do ban co de dados que requerem a execução de operações posteriores, o conceito de evento/partida de THM pode ser usado. Um evento ou depende de uma operação ou de uma condição de tempo. Uma partida (trigger) descreve as ações que devem ser levadas a cabo se ocorrer o ou/os evento(s) associado(s). Finalmente para garantir a integridade semântica do banco de da

dos relativamente à hierarquia das classes e também relativo às relações e suas inversas, são introduzidos em THM os chamados efeitos colaterais. Eles descrevem as operações que devem ser executadas como uma consequência de se ter executado outras operações.

3. O MODELO DE REQUISITOS

A primeira fase da metodologia de projeto THM pretende derivar o Modelo de Requisitos(MR), que pode ser considerado como uma primeira formalização do SI a ser desenvolvido. O MR será derivado de uma descrição informal dos requisitos do SI usando diversos ciclos de iteração como mostrado na seção 3.3. De nosso ponto de vista é importante considerar tanto os aspectos dinâmicos quanto os estáticos de um SI desde o início uma vez que o projeto do esquema estrutural será imediatamente influenciado pelos aspectos dinâmicos do SI e vice-versa. Portanto, o MR compreende dois tipos diferentes de grafos para cobrir os aspectos estáticos e dinâmicos. Uma vez que o MR será usado na segunda fase do projeto como a base para derivar a rede THM, os conceitos de modelagem oferecidos no MR são orientados para os conceitos de modelagem estáticos e dinâmicos definidos em THM, o modelo semântico de dados que será usado finalmente para representar o esquema conceitual do SI.

O MR é composto de dois grafos o Grafo de Comportamento(GC) e o Grafo Estrutural(GE). Posteriormente, discute-se em detalhes os conceitos de modelagem providos por estas estruturas de grafos.

3.1. O Grafo de Compostamento(GC)

O GC provê uma descrição funcional do SI, especificando o fluxo de informações entre funções e as chamadas unidades de informação (compare com SADT ([13])). Adicionalmente, o GC descreve o comportamento de cada função por meio das pré e pós-condições, introduz condições de ocorrência para eventos e especifica restrições de integridade para as unidades de informação.

Seu propósito é prover uma visão das funções necessárias no SI e suas semânticas e das informações manipuladas pelas funções.

Consideremos a definição do GC:

Definição 3.1:

Um Grafo de Comportamento(GC) é definido como segue:

Um GC é uma rede dirigida (F, I, E, IF) onde:

- (1) F é o conjunto das funções,
- (2) I é o conjunto de unidades de informação,
- (3) E é o conjunto de eventos,
- (4) $IF \subseteq ((Ix F) \cup (Fx I) \cup ((Ex F) \cup (Fx E)))$ é uma relação representando os arcos do GC.

(1) Funções:

O conjunto de funções F é composto de dois subconjuntos disjuntos: $F = EF \cup CF$, onde

(i) EF é o conjunto das funções elementares,

(ii) CF é o conjunto das funções complexas. Uma função complexa é composta de funções elementares e/ou outras funções complexas. Ao introduzir o conceito de funções elementares e complexas, um GC pode ser decomposto em uma hierarquia de diagramas, i. e., uma função complexa de nível i pode ser representada por um GC separado de nível $i+1$.

(2) Unidades de Informação:

O conjunto de Unidades de informação I é composto de dois subconjuntos disjuntos:

$I = EXI \cup INI$, onde

(i) EXI é o conjunto de unidades externas de informação descrevendo informações que não serão mantidas no SI, i. e., entrada para o SI e saída do SI.

(ii) INI é o conjunto de unidades internas de informação que serão parte do SI e posteriormente transformadas em classe de entidades na descrição do esquema conceitual. Para uma dada função $f \in F$ o conjunto de unidades de informação de entrada é definido por $pre(f) = \{i | (i, f) \in ((IXF) \cap IF)\}$, o conjunto das unidades de informação de saída por: $pos(f) = \{i | (f, i) \in ((FXI) \cap IF)\}$.

(3) Eventos:

O conjunto de eventos é composto de dois subconjuntos disjuntos: $E = EE \cup IE$, onde

(i) EE é o conjunto de eventos externos que representam as trocas de estado do mundo real que resultam em uma troca de estado do SI. A ocorrência dos eventos externos é dependente das condições fora do escopo formal do SI.

(ii) IE é o conjunto dos eventos internos. Um evento interno causa uma troca de estado do SI e sua ocorrência depende ou de uma outra troca de estado (gerada por uma função) ou de um sistema manuseado por condições de tempo.

(4) Arcos:

Arcos conectando funções e unidades de informação especificam o fluxo de informações no GC. Um arco, que é um elemento de (ExF) , conecta uma função com o evento que especifica sua condição de ativação. Um arco que é um elemento de (FxE) , representa o fato de que o evento associado com a função será ativado ao executar a função.

(5) Inscrições:

Unidades de informação, eventos e atividades podem ser inscritos ou por nomes, descrevendo informalmente o significado dos diferentes elementos do GC, ou por um texto descrevendo as condições que devem ser atingidas por eles. Condições para unidades de informação e eventos são chamadas de restrições e ou especificam restrições para as entidades das unidades de informação ou descrevem condições de ocorrência para eventos. Condições para atividades são descritas por pré e pós-condições, respectivamente, especificando condições para as entidades das unidades de informação de entrada e das

unidades de informação de saída, respectivamente. As condições são coletadas em uma lista de condições que é referenciada a partir das funções correspondentes, unidades de informação, e eventos do GC.

(6) Comportamento:

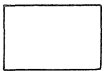
Uma função $f \in F$ consome/ler elementos (entidades) de (alguma de) suas unidades de informação de entrada e entrega elementos (entidades) para (alguma de) suas unidades de informação de saída. As restrições que devem ser mantidas para as entidades das unidades de informação de entrada (saída) são especificadas pelas pre-/pós-condições da função.

Se uma função $f \in F$ está associada com um evento por um arco que é um elemento de $(E \times F)$, a ocorrência do evento é um pre-requisito para a execução da função. Se um evento está conectado a uma função $f \in F$ por um arco que é um elemento de $(F \times E)$ o evento é gerado tão logo a função tenha sido executada.

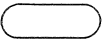
Do mesmo modo que com uma rede canal/agência, o GC não provê uma especificação única com relação ao consumo e geração de entidades durante a execução de uma função:

- (i) nem todas as unidades de informação de entrada devem fornecer entidades;
- (ii) nem todas as unidades de informação de saída recebem unidades;
- (iii) os arcos entre unidades de informação e funções não especificam se as entidades são consumidas, modificadas ou simplesmente lidas;
- (iv) os arcos não especificam o número de entidades que são consumidas ou produzidas.

Na construção do GC será usada a seguinte notação:



Para funções.



Para unidades internas de informação.



Para unidades externas de informação.



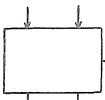
Para eventos.



Para arcos (fluxo de informações).



Para entrada ou saída de informações mutuamente exclusivas.



Um retângulo envolvendo funções, eventos..., para representar o detalhamento de uma função "A".

GC:função "A"

3.2. O Grafo Estrutural (GE)

Para modelar os aspectos estáticos de um SI o GE é usado no MR. O GE baseia-se em um modelo semântico de dados oferecendo um subconjunto dos conceitos de modelagem definidos em THM. Uma vez que THM será usado para a descrição do esquema conceitual (na fase 2), o processo de análise efetuado na primeira fase é dirigido para os conceitos finais de modelagem desde o início. A principal restrição é que os princípios de abstração de THM (generalização, agregação, agrupamento) não foram incluídos no GE. A razão para tal restrição do modelo semântico de dados na primeira fase do projeto é que o especialista de aplicação não está familiarizado com todos os diferentes conceitos de modelagem usualmente oferecidos por tal modelo. Além disso não é prático desenvolver uma descrição de um esquema conceitual completo em uma fase de projeto.

O GE é desenvolvido através da análise das unidades de informação do GC como também das restrições e pré-/pós-condições definidas no GC. Durante este processo de derivação são definidas as classes de entidades e são estabelecidas as relações entre as classes de entidades. As restrições requeridas para as classes de entidades do GE são adicionadas à lista de condições já introduzidas no GC.

A seguir é apresentada uma definição do GE, i. e., o modelo semântico de dados restrito. Entretanto, não será apresentada uma DDL uma vez que pode ser usado diretamente o subconjunto de THM/DDL ([14] e [15]) requerido para nosso modelo semântico de dados restrito.

Definição 3.2.

Um Grafo Estrutural (GE) é definido como segue:

Um GE é um grafo dirigido (CE;R)

(i) CE é o conjunto de classes de entidades,

(ii) R é o conjunto de relações binárias entre classes de entidades representando os arcos do GE.

(1) Classes de Entidades:

Uma entidade representa um objeto do mundo real. Um conjunto de entidades que têm as mesmas propriedades é uma classe de entidades.

(2) Relações:

O conjunto das relações é composto de três subconjuntos disjuntos:

$R = CCR \cup CMR \cup MMR$, onde:

(i) CCR é o conjunto de relações classe-a-classe,

(ii) CMR é o conjunto de relações classe-membro,

(iii) MMR é o conjunto de relações membro-a-membro:

Para representar uma relação r entre duas classes $C1$ e $C2$ é usada a notação ' $C1 \ r \ C2$ '.

(3) Restrições:

- (i) Todas as classes de entidades devem ter nomes únicos;
- (ii) Todas as relações devem ter nomes diferentes;
- (iii) A especificação de relações classe-a-membro e membro-a-membro deve incluir uma especificação de cardinalidade $c = (mim, max)$. Para uma dada relação entre duas classes A e B (mim, max) especifica o número mínimo e o número máximo de entidades da classe B relacionadas a uma entidade de A. ('*' representa um número indeterminado, 'L' representa um limite previamente estabelecido).

É oferecido um conjunto de relações predefinidas que pode ser usado posteriormente pa
ra descrever as hierarquias de abstração requeridas para a descrição do esquema con
ceitual.

A seguir, são apresentadas as relações "default" para relações membro-a-membro.

Relações membro-a-membro:

Relação	Cardinalidade	Descrição
1) C1 é um C2	(1,1)	.Todo elemento de C1 tem que ser elemento de C2.
C1 é um C2	(0,1)	.Um elemento de C1 pode ser um elemento de C2.
2) C1 é parte de C2	(1,1)	.Um elemento X de C1 é sempre parte de exa tamente um elemento Y de C2.
C1 é parte de C2	(0,1)	.Um elemento X de C1 pode ser parte de um elemento Y de C2.
(Para $max > 1$ tem-se casos análogos).		
3) C2 tem-comp. C1	(1,1)	.Um elemento Y de C2 sempre inclui exata mente um componente X de C1.
(Para $max > 1$ tem-se situação análoga 'max = *' e também 'min = 0' não devem ocorrer).		
4) C1 é elemento de C2	(1,1)	.Uma entidade X de C1 é sempre um elemen to de exatamente uma entidade de C2.
(Todos os outros tipos de cardinalidade são também permitidos).		
5) C2 tem-elemento C1	(1,n)	.Uma entidade Y de C2 tem n entidades X1, X2...,Xn de C1 como elementos.
('min = 0' é permitido. 'n>1' tem que ser mantido)		

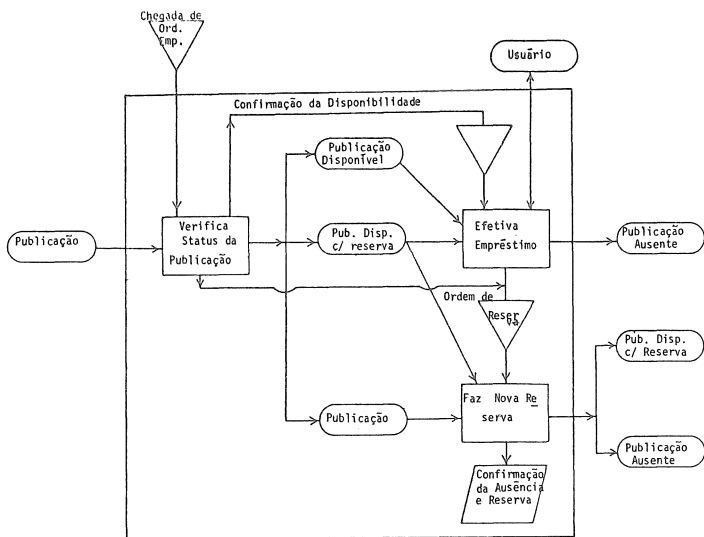


Figura 2d: GC - Fluxo de Informação de Verifica_Disponibilidade

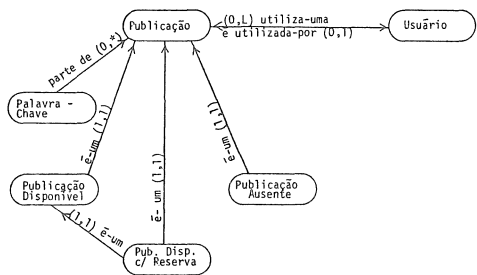


Figura 3: Grafo Estrutural

FUNÇÃO	CLASSE/RELAÇÃO		
	CONSULTA	ATUALIZA SITUAÇÃO	EFETIVA EMPRÉSTIMO
Publicação	L		
Usuário		M	*
Publicação Ausente			G
Publicação Disponível		G	C
Publicação Disp. c/ Reserva		G	C
Palavra Chave	L		

Figura 4: Parte da Tabela de Acessos para o GC da Figura 2.

3.3. Os passos para desenvolver MR

Como foi mostrado na introdução a metodologia de projeto é dividida em duas fases. Nesta seção serão discutidos os vários passos que devem ser efetuados durante a primeira fase para o desenvolvimento do MR. No final desta seção serão ilustrados os diferentes passos com a discussão do problema de controle do acervo bibliográfico de uma pequena biblioteca. A suposição básica é que não será assumida uma estratégia estritamente Top-down. Ao contrário, assume-se que as estruturas dos grafos do MR são desenvolvidas de maneira iterativa, i.e., modificar uma das estruturas de grafo pode resultar em modificações de outras estruturas de grafo.

Passo 1: O Desenvolvimento do GC

Dependendo dos passos anteriores, os seguintes casos podem ser distinguidos durante o desenvolvimento do GC:

- a) É desenvolvida uma primeira visão do universo de discurso, analisando o fluxo de informações entre funções e determinando os eventos que ativam as funções.
- b) Um GC existente é estendido com a inclusão de novas funções, unidades de informação, e eventos.
- c) Unidades de informação de entrada/saída que ainda não foram consideradas são conectadas com as funções correspondentes.
- d) Uma função complexa é decomposta em diversas subfunções conectados por arcos de fluxo de informação. Este processo de decomposição é continuado enquanto a análise do fluxo de informações resulte na identificação de novas subfunções (veja figura 2).
- e) As atividades do GC são analisadas relativamente às pré/pós-condições que têm que existir para uma execução satisfatória das atividades. Ou as atividades são inscritas pelas pré/pós-condições ou elas são inseridas na lista de condições (figura 5) que é então referenciada no GC.
- f) As unidades de informação são especificadas posteriormente com a inserção de restrições na lista de condições para garantir a integridade semântica das informações mantidas nestas unidades (veja figura 5).
- g) A descrição dos eventos é desenvolvida posteriormente com a especificação das condições de ocorrência para os eventos. Estas condições de ocorrências são também registradas na lista de condições.

Passo 2: A Derivação do Grafo Estrutural(GE)

Ao derivar o GE as unidades de informação e também a lista de condições do GC devem ser analisadas. Deve ser verificado como as informações descritas pelas unidades de informação e as restrições do GC podem ser representadas por classes de entidades e relações. Adicionalmente, deve ser verificado que restrições da lista de condições são cobertas pelas relações definidas entre as classes de entidades. Se é introduzida uma nova relação, tem

FUNÇÃO	PRÉ/PÓS-CONDIÇÕES	
CATALOGAÇÃO DE DOCUMENTOS.	Pré-Cond. Pós-Cond.	Existência ou destruição de um documento.
CONSULTA	Pré-Cond. Pós-Cond.	
EMPRÉSTIMO	Pré-Cond. Pós-Cond.	Existência da publicação.
REGISTRO	Pré-Cond. Pós-Cond.	Registro de documento não existente ou destruição de publicação que já existe. Ter sido efetuado o registro da publicação (cadastramento ou cancelamento).
ATUALIZAÇÃO SITUAÇÃO	Pré-Cond. Pós-Cond.	A publicação fica disponível ou reservada e o total de empréstimos do usuário é atualizado.
VERIFICA REGULARIDADE	Pré-Cond. Pós-Cond.	Existência do usuário.
INCLUI NOVO USUÁRIO	Pré-Cond. Pós-Cond.	Não existência do usuário. Ter sido efetuada a inclusão do novo usuário.
NOTIFICA USUÁRIO	Pré-Cond. Pós-Cond.	Usuário em débito ou limite atingido.
VERIFICA DISPONIBILIDADE.	Pré-Cond. Pós-Cond.	
VERIFICA STATUS DA PUBLICAÇÃO	Pré-Cond. Pós-Cond.	
EFETIVA EMPRÉSTIMO	Pré-Cond. Pós-Cond.	Disponibilidade da publicação para o usuário. Não disponibilidade da publicação se for última cópia e total de empréstimo do usuário atualizado.
FAZ NOVA RESERVA	Pré-Cond. Pós-Cond.	Não disponibilidade da publicação. Ter sido efetuada a reserva de uma publicação.

UNIDADE DE INFORMAÇÃO	RESTRIÇÕES
Pub. Ausente	Não é uma Publicação Disponível
Pub. Disponível	Não é uma Publicação Ausente
Pub. Disponível c/Reserva	Não é uma Publicação Ausente

Fig. 5: Lista de Condições

que ser verificado se pode ser usada, uma das relações "default" (compare com a seção 3.2). O passo 2 pode ser dividido nos seguintes subpassos:

- a) Para cada unidade de informação do GC que foi introduzida durante o último ciclo de iteração, deve ser examinado se pode ser representada por novas relações entre as classes de entidade existentes ou pela introdução de uma nova classe.
- b) O GC é analisado posteriormente para identificar relações adicionais que são requeridas no GE para expressar restrições que devem ser mantidas entre as entidades (classes de entidades). Esta análise é orientada inicialmente para relações que podem ser derivadas das entidades de unidades de informação de entrada/saída de uma função.
- c) O GE é examinado em relação a diversos critérios de modelagem que serão estabelecidos pelo projetista do GC.

Exemplos de tais critérios:

- As duas relações 'C1 é parte de C2' e 'C1 é elemento C2' não devem coexistir entre C1 e C2.
- Se é definida uma relação 'C1 é um C2' com cardinalidade (1,1) então não é permitida uma segunda relação 'C2 é um C1' com cardinalidade (1,1).
- Se existem duas relações 'C1 é um C2' com cardinalidade (1,1) e 'C2 é um C1' com cardinalidade (0,1), então de uma relação 'C2xC3' pode ser deduzida uma relação 'C1xC3' que portanto não precisa ser incluída no GE (veja figura 3).

Após a definição do GE obtém-se uma primeira visão do SI.

Passo 3:

A tabela de acessos é construída de modo a especificar para cada classe de entidade e para cada relação do GE a maneira de sua manipulação pelas funções do GC. Esta análise é requerida pois os arcos do GC não especificam se as entidades de uma unidade de informação são lidas, consumidas, modificadas ou geradas. Entretanto, esta espécie de informação é essencial para a posterior derivação da rede THM.

A tabela de acessos é organizada como uma matriz com as linhas representando as classes de entidades e as relações e com as colunas representando as funções. Um elemento da matriz é preenchido com G/C/L/M se uma função correspondentemente gera/consome/ler/modifica a classe de entidade ou relação. (Veja figura 4). Se uma classe ou uma relação não tem uma entrada na tabela de acessos especificando a espécie de acesso para tal, é requerido um novo ciclo de iteração para introduzir as funções e eventos correspondentes.

Passo 4: Análise das Funções

Para determinar se uma função está definida exatamente, deve ser verificado se:

- Existem situações em que nem todas as unidades de informação de entrada/saída são acessadas pela função,

- as pré/pós-condições especificadas para a função cobrem alternativas que devem ser representadas por diversas subfunções.

Tendo identificado tais situações, cada uma das transições de estado alternativas que são representadas por uma função, são representadas por uma subfunção correspondente. Geralmente, durante este processo de decomposição, as regras para morfismos de redes não devem ser violadas (Veja [4]).

0 exemplo de automação do controle do acervo bibliográfico de uma biblioteca:

Ao longo do texto foi apresentado, sem maiores detalhes, o sistema de controle do acervo bibliográfico de uma pequena biblioteca. Na figura 2a apresentamos uma visão global do sistema. As funções "Catalogação de Documentos" e "Empréstimo" foram decompostas em um GC de segundo nível (figura 2b e 2c, respectivamente). A figura 2d é uma decomposição (detalhamento) da função "verifica disponibilidade" que foi apresentada no segundo nível (figura 2c) sendo, portanto, um GC de terceiro nível. "Documento", "Resposta-a-consulta" e "Relatório de publicações destruídas" são exemplos de unidades externas de informação. "Publicação" e "Usuário" são exemplos de unidades internas de informação. "Chegada de documento" e "Chegada de ordem de destruição" são exemplos de eventos (externos) de ativação de uma função e a ocorrência de qualquer dos dois irá ativar a função "Registro".

Na figura 5 são encontradas as pré/pós-condições e restrições descritas na seção 3.3. A figura 3 mostra o Grafo Estrutural correspondente ao GC da figura 2.

Conclusão

Foi apresentado um caso prático de desenvolvimento do esquema conceitual para um sistema de informações usando uma nova metodologia de projeto. A metodologia usada é bastante promissora por estabelecer uma boa base para comunicação entre o especialista de aplicações e o ABD já que o Modelo de Requisitos desenvolvido na primeira fase é orientado para o especialista de aplicações enquanto que a rede THM a ser desenvolvida na segunda fase é orientada para conceitos de banco de dados.

Referências

- [1] Brodie, M.L.: Association: A Data Base Abstraction for Semantic Modeling, em Chen(ed): Entity-Relationship Approach to Information Modeling and Analysis, 1981.
- [2] Bracchi, C.; Pernici, B.: The Design Requirements of Office Systems, em: ACM Trans. on Office Information Systems 2,2 (Abril 1984), 151-170.
- [3] Genrich H.J.; Lautenbach, K.: System Modelling with High-Level Petri Nets, em: Theoretical Computer Science 13(1981), 109-136.
- [4] Genrich H.J.; Stankiewicz-Wiechno, E.: A dictionary of Some Basic Notations in Net Theory, em: Brauer(ed.): Net Theory and Application, LNCS 84, Springer Verlag, 1980.

- [5] Gustafsson, M.R. et al.: A Declarative Approach to Conceptual Information Modelling, em: Olle et al(eds.): Information Systems Design Methodologies: A Comparative Review, North-Holland Publ. Co., 1982.
- [6] Hammer, M.; McLeod, D.: Data Base Description with SDM: A Semantic Data Base Model, em ACM TODS 6,3 (1981).
- [7] Horndasch, A.; Studer, R.; Yasdi, R.: An Approach to Conceptual Schema Design of Information Systems, em Proc. IFIP WG 8.1 Working Conference on Theoretical and Formal Aspects on Information Systems, A. Sernadas e A. Olivé (eds.) Sitges, North-Holland Publ. Co., 1985.
- [8] Lundeberg, M.: The ISAC Approach to Specification of Information Systems and its Application to the Organization on an IFIF Working Conference, em: Olle et al (eds.): Information Systems Design Methodologies: A Comparative Review, North-Holland Publ. Co., 1982.
- [9] Olle, T.W. et (eds.): Information Systems Design Methodologies: A Comparative Review, North-Holland Publ. Co., 1982.
- [10] Richter, G.: IML - Inscribed Nets for Modelling Text Processing and Data (Base) Management Systems, em Proc. 7th Int. Conf. on Very Large Data Bases, Cannes, 1981.
- [11] Richter, G.: Utilization of Data Access and Manipulation in Conceptual Schema Definitions, em: Information Systems 6, (1981), 53-71.
- [12] Richter, G.; Durchholz, R.: IML - Inscribed High-Level Petri Nets, em: Methodologies: A Comparative Review, North-Holland Publ. Co., 1982.
- [13] Ross, D.T.: Structured Analysis: A Language for Communicating Ideas, em: IEEE Trans. on Software Engineering 3,1 (1977).
- [14] Schiel, U.: An Abstract Introduction to the Temporal-Hierarchic Data Model (THM), em: Proc. 9th Int. Conf. on Very Large Data Bases, Florence, 1983.
- [15] Schiel, U.: A Semantic Data Model for Conceptual Schemas and their Mapping to Internal Relational Schemas (en Alemão), University of Stuttgart, Doctoral Thesis, 1984.