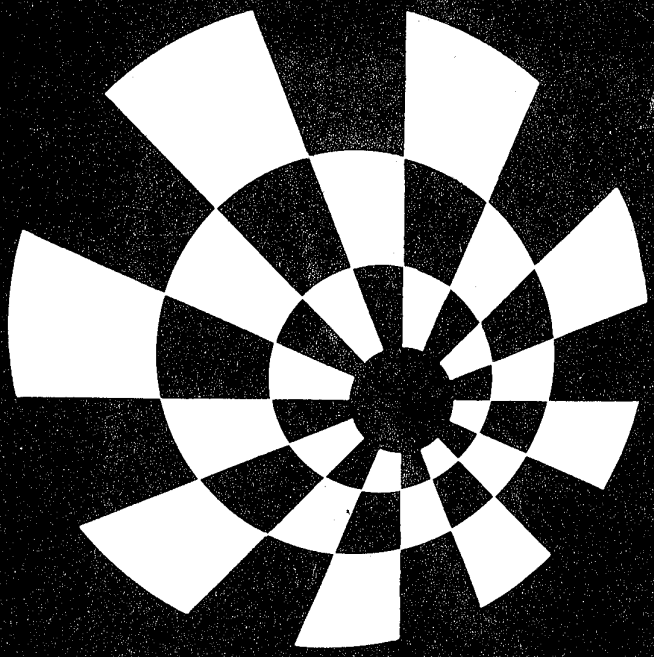


3º SIMPÓSIO BRASILEIRO de BANCO de DADOS

23.25 de março de 1988
Recife – Pernambuco

ANAIS

UM PROTÓTIPO DE FERRAMENTA PARA PROJETO/REPROJETO DE BANCOS DE DADOS

Galeno J. de Sena
Faculdade de Engenharia de Guaratinguetá

Antônio L. Furtado
Pontifícia Universidade Católica do Rio de Janeiro

Sumário

Uma ferramenta especializada para projeto/reprojeto de bancos de dados é apresentada. A ferramenta suporta uma disciplina de modularização desenvolvida para ajudar no projeto e manutenção de esquemas de bancos de dados complexos. Na fase de projeto, a ferramenta invoca um sistema de geração de planos para a auxiliar na verificação de alguns dos requisitos semânticos da metodologia.

1. Introdução

O que motiva, basicamente, o desenvolvimento de uma disciplina de modularização [CFT,FCT], a ser usada na especificação de esquemas de bancos de dados, e a construção de uma ferramenta suportando a sua aplicação, é o problema de organização do projeto e manutenção de esquemas de bancos de dados complexos, considerado no sentido amplo de como se descrever as estruturas de dados (parte estática) e as transações (parte dinâmica) de um sistema de informação (SI) baseado em um banco de dados. A utilidade de ferramentas [BGM,CFT,Ce,FCT] para auxiliar no projeto de sistemas de informação [LM] baseados em bancos de dados é largamente reconhecida, entre outras razões elas podendo constituir o único modo efetivo de se conduzir métodos de projeto formais para o domínio das aplicações. Sistemas especialistas [BGM,FM,Ni], incorporando o conhecimento de especialistas humanos em alguma área, estão sendo aplicados na solução de um grande número de problemas. Uma combinação dos dois conceitos - ferramentas de projeto e sistemas especialistas, daí resultando as ferramentas especializadas [BGM,CFT,Ce,FCT] - constitui, como se poderia esperar, uma área de pesquisa intensa [BGM].

O conhecimento a incorporar na ferramenta deve refletir, sobretudo, um método para projeto/reprojeto de bancos de dados [FCT]. A capacidade de inferência [FM,Ko,Ni,Wa] pode, em especial, usar regras gerais para sintetizar seqüências de aplicações de funções (ativação de um sistema de geração de planos), capazes de conduzir a aplicação do estado corrente a um estado caracterizado (parcialmente) por uma conjunção de fatos dada.

O propósito deste artigo é apresentar uma ferramenta para projeto/reprojeto de bancos de dados [CFT,FCT,Se], implementando uma disciplina de modularização, descrita em [CFT], e ativando um sistema de geração de planos, baseado na ferramenta de [FM,VF]. O texto está organizado como segue. A secção 2 apresenta os conceitos básicos concernentes à disciplina de modularização. A secção 3 descreve o protótipo de geração de planos invocado pela

ferramenta de projeto. O protótipo de projeto/reprojeto é descrito na secção 4. Exemplos de uso do protótipo são apresentados na secção 5.

2. A disciplina de modularização

A fase de projeto da metodologia é caracterizada pela adição sistemática de módulos, de modo estruturado, a um núcleo de banco de dados, possivelmente vazio, definindo-se, a nível de módulo, os conjuntos de objetos compatíveis com o seu tipo. Uma vez que o projetista do banco de dados pode ter a sua percepção da aplicação modificada ou que esta pode evoluir, conclui-se que o procedimento de especificação de esquemas de bancos de dados deve ter a natureza iterativa, permitindo o retrocesso e a alteração na definição de módulos, o que caracteriza a fase de reprojeto da metodologia.

Há no método três tipos de módulos: primitivo, subordinante e externo. O ABD escolhe o tipo primitivo para descrever porções da aplicação que têm sentido por si mesmas. Módulos subordinantes são construídos a partir de módulos já especificados, os módulos subordinados, o construtor de subordinação sendo usado pelo ABD quando desejar introduzir, entre outros objetos, uma restrição de integridade que pode ser violada por alguma operação definida previamente em algum módulo do esquema, a operação tornando-se escondida e o módulo que a contém, subordinado. Módulos externos são também definidos sobre módulos previamente introduzidos, os módulos estendidos, o ABD devendo fazer uso deles quando desejar introduzir visões e operações sobre visões. Um módulo é conceitual se ele é primitivo ou subordinante. Um esquema de banco de dados modular é caracterizado por um conjunto de módulos, cada módulo sendo um módulo conceitual ou externo.

Fazendo uso do modelo de dados relacional [Da], um módulo pode conter, não importando o seu tipo, definições de relações (tabelas), restrições de integridade e operações. As restrições de integridade estabelecem quais os estados que são válidos no banco de dados (restrição estática) e que transições de estados são válidas (restrição de transição). Para especificar, num módulo conceitual, que o corpo de uma operação contém testes para garantir a não violação de uma restrição de integridade, introduz-se uma cláusula de garantia no esquema, que corresponde a uma declaração da forma:

"operação garante restrição de integridade"

os objetos participando do relacionamento devendo ter sido introduzidos no mesmo módulo. Para o caso específico de módulos subordinantes deve haver a especificação de cláusulas de reconhecimento. Para especificar que a aplicação de uma operação pode violar uma restrição de integridade, introduz-se uma destas cláusulas, a qual corresponde a uma declaração da forma:

"operação pode-violar restrição de integridade"

os objetos participando do relacionamento devendo ter sido especificados em módulos diferentes: a operação em um módulo subor-

dinado e a restrição de integridade no módulo subordinante.

Em módulos externos dever haver a introdução de mapeamentos de definição de visão e de procedimentos substitutivos. Um mapeamento de definição de visão, introduzido para cada relação do módulo, permite defini-la em termos das relações dos módulos estendidos. Um procedimento substitutivo, especificado para cada operação do módulo, é uma operação definida sobre a união das relações dos módulos estendidos, devendo ser uma tradução exata e aceitável da operação correspondente do módulo externo [FC].

Uma das características da metodologia é a existência de uma ordem natural [FCT] para criar ou modificar uma especificação. A nível de módulos, se o módulo M subordina ou estende o módulo M', então M' deve ser definido antes de M. A nível de componentes a ordem é: (1) relações, (2) restrições de integridade, (3) cláusulas de recobrimento, (4) operações, (5) cláusulas de garantia. A ordem natural caracteriza a ordem de definição [CFT] da metodologia, na qual o ABD deve definir ou modificar os objetos do esquema.

A especificação de um esquema de banco de dados com o uso dos construtores da metodologia não garante que o banco de dados estará sempre num estado consistente, mesmo que as modificações dos usuários sejam efetuadas somente com as operações que lhes sejam visíveis. Além disso, nada impede que uma relação permaneça sempre vazia, pela ausência de uma operação de inserção apropriada, e que uma operação se torne inativa, devido a seu recobrimento e à não introdução de uma nova operação que a chame no esquema. Estes problemas são evitados adotando-se um conjunto de requisitos de projeto, que definem o que é um projeto consistente e formam o núcleo da disciplina de modularização [CFT].

A formalização dos requisitos de projeto gera as leis de integridade, que são implementadas no protótipo para constituir um módulo de verificação dos requisitos nas fases de projeto e reprojeto do esquema. A formalização dos requisitos usados pela ferramenta de reprojeto inclui as leis de propagação. Estas são usadas para otimizar o reprojeto por dirigir a atenção para aqueles componentes que podem ter sido afetados por alguma alteração efetuada pelo ABD e que podem, como conseqüência, necessitar também de alguma alteração por sua vez. A ordem de definição garante que a propagação de alterações somente ocorre na direção ascendente (ou seja, do objeto que sofreu alteração para objetos definidos depois dele segundo ordem de definição).

3. O protótipo de geração de planos

O problema de formação de planos [Ko] é caracterizado por um estado inicial, um estado objetivo e um conjunto de ações utilizadas nas transições de estados da aplicação. Busca-se um plano - uma seqüência apropriada de ações capaz de transformar o estado inicial num estado final que satisfaça a descrição do estado objetivo (este será um subconjunto do estado final). De um modo geral uma ação é definida pela especificaçao de suas pré-condições - enunciados que devem ser válidos em um estado antes que a ação possa ser executada, e pós-condições - enunciados que são válidos no estado resultante da aplicação da ação.

Pós-condições podem ser de dois tipos: (1) novos enunciados adicionados à especificação do banco de dados e (2) enunciados que são válidos no estado anterior e que continuam válidos no estado resultante da aplicação da ação. O problema de como especificar os enunciados do tipo de (2) é usualmente chamado de "frame problem" [Ko,Ni,Wa].

A implementação do protótipo, efetuada em micro-PROLOG [CM], está baseada no sistema WARPLAN [Wa] - um gerador de planos de propósito geral para domínios descritos em um formalismo próximo daquele usado pelo sistema STRIPS [Ni], e no protótipo apresentado nos trabalhos de [FM,VF]. O protótipo é composto de dois módulos: um específico, o módulo de aplicação [FM], que se refere a um particular SI [LM] de bancos de dados e corresponde à especificação do esquema conceitual da aplicação, e outro, geral, o módulo de processamento [FM], que permite a execução de experimentos com fatos do banco de dados, a nível do esquema conceitual da aplicação.

O módulo de aplicação é estruturado com a introdução de cláusulas para os predicados seguintes:

"operation-on":	para especificação dos domínios de onde são obtidos os valores dos parâmetros das ações;
"added":	para especificação dos fatos (enunciados) adicionados pelas ações;
"deleted":	para especificação dos fatos excluídos pelas ações;
"conditions":	para especificação das pré-condições para a aplicação de ações e
"imposs":	para especificação de conjunções de fatos que violam as restrições de integridade do banco de dados.

Os axiomas do módulo de processamento do protótipo adaptam um algorítmo [Wa] para a geração de seqüências alternativas de aplicações de funções (ações) que transformam a especificação de uma dada aplicação de um estado inicial (ou de qualquer estado satisfazendo a descrição inicial) para um estado objetivo (ou algum estado satisfazendo a descrição do objetivo).

4. O protótipo de projeto/reprojeto

O protótipo resultante da implementação do método de projeto/reprojeto e da integração deste com o de geração de planos está organizado na forma de duas unidades [CFT,FCT,Se]: uma ferramenta de projeto - auxiliando o ABD na especificação de novos módulos, e uma ferramenta de reprojeto - ajudando-o na especificação de alterações para o esquema modular. A integração da ferramenta de projeto com o protótipo de geração de planos foi realizada por meio de um programa de ligação [Se]. O protótipo foi implementado em micro-PROLOG [CM] estendido com o sistema APES [HS]. A organização das ferramentas é analisada a seguir.

4.1 A ferramenta de projeto

A figura 1 mostra os componentes da ferramenta de projeto e o fluxo de controle, uma seta simples indicando a ativação de um componente e uma seta dupla, a saída de resultados.

Os componentes têm as funções seguintes:

(a) o gerenciador de projeto (GPR) rege a criação de módulos e de seus objetos em uma ordem compatível com a ordem de definição;

(b) o administrador de diálogos (AD) controla a comunicação com o ABD, verifica a correção de uma resposta e armazena, no dicionário de dados, cláusulas descrevendo os objetos do esquema modular;

(c) o dicionário de dados (DD) contém a descrição do esquema modular na forma de cláusulas;

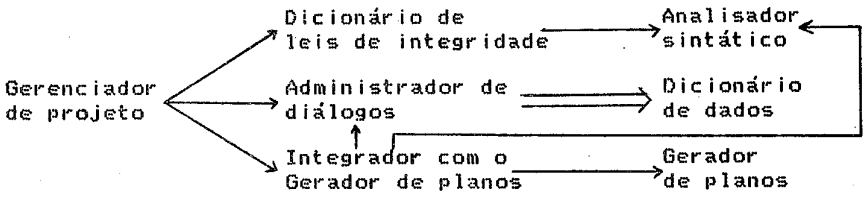
(d) o analisador sintático (AS) verifica a sintaxe livre do contexto e constrói as árvores de análise para fórmulas bem formadas do cálculo de predicados de primeira ordem e para operações da linguagem de programação formal descrita em [CB];

(e) o dicionário de leis de integridade (DLI) contém uma implementação das leis de integridade da metodologia;

(f) o integrador com o gerador de planos (IGP) gera as cláusulas do módulo de aplicação do protótipo de geração de planos e

(g) o gerador de planos (GPL) contém o módulo de processamento do protótipo de geração de planos e, usando as cláusulas do módulo de aplicação, verifica a suficiência das pré-condições das operações do esquema.

Figura 1: Componentes e fluxo de controle na ferramenta de projeto



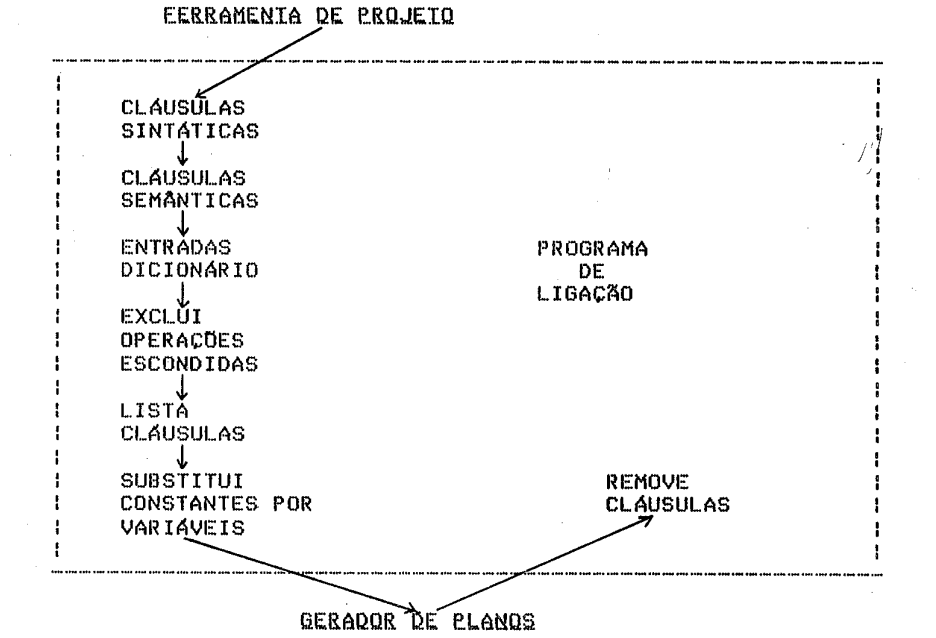
A ferramenta não implementa diretamente os construtores da metodologia: ela estabelece um diálogo com o ABD do qual resulta a descrição de novos módulos. Um esquema modular é armazenado no DD na forma de um conjunto de cláusulas definindo um predicado designado por "tab", que toma parte então no programa lógico que define a ferramenta.

4.2 Integração da ferramenta de projeto com o módulo de geração de planos

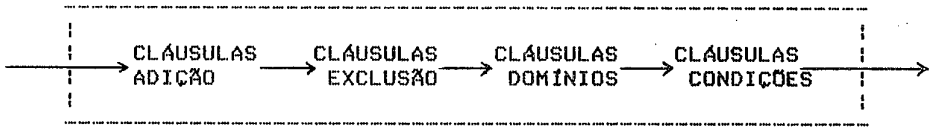
Foi mencionado que a integração da ferramenta de projeto com o gerador de planos foi realizada através de um programa de ligação [Se], este correspondendo ao componente IGP da figura 1. Este estrutura o módulo de aplicação para o gerador de planos a partir das operações e das restrições de integridade do módulo considerado. A estrutura desse programa de ligação é apresentada na figura 2, as setas indicando a ordem de ativação de cada uma das funções que o constituem.

O propósito de cada uma das funções do programa de ligação é

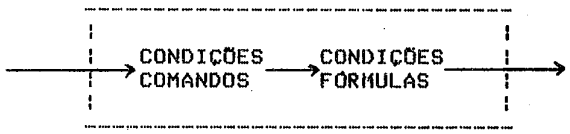
Figura 2: Estrutura do programa de ligação da ferramenta de projeto com o gerador de planos



FUNÇÃO CLÁUSULAS SINTÁTICAS



FUNÇÃO CLÁUSULAS CONDIÇÕES



apresentado a seguir:

CLÁUSULAS SINTÁTICAS: abarca as subfunções seguintes:

CLÁUSULAS ADIÇÃO: gera as cláusulas do predicado "added";

CLÁUSULAS EXCLUSÃO: gera as cláusulas do predicado "deleted";

CLÁUSULAS DOMÍNIOS: gera as cláusulas do predicado 'operation-on';

CLÁUSULAS CONDIÇÕES: comporta as subfunções seguintes:

CONDIÇÕES COMANDOS: estrutura as listas de condições das cláusulas do predicado 'conditions';

CONDIÇÕES FÓRMULAS: identifica os elementos dessas listas de condições.

CLÁUSULAS SEMÂNTICAS: constrói as cláusulas do predicado 'imposs' mediante diálogo com o ABD.

ENTRADAS DICIONÁRIO: gera entradas do dicionário de relações[CM] para os predicados do módulo de aplicação.

EXCLUI OPERAÇÕES ESCONDIDAS: exclui do módulo de aplicação as cláusulas de operações escondidas, introduzindo cláusulas semanticamente equivalentes, definidas sobre as operações que chamam essas operações escondidas.

LISTA CLÁUSULAS: exibe o módulo de aplicação construído.

SUBSTITUI CONSTANTES POR VARIÁVEIS: nesta, o ABD, fazendo uso das facilidades de edição do micro-PROLOG, substitui convenientemente constantes por variáveis nas cláusulas dos predicados do módulo de aplicação.

REMOVE CLÁUSULAS: responsável pela exclusão do programa, após retornar do gerador de planos, das cláusulas dos predicados do módulo de aplicação.

O IGP, após a execução da função "SUBSTITUI CONSTANTES POR VARIÁVEIS", ativa o GPL, o qual procura então encontrar seqüências (alternativas) de aplicações de operações (ações) capazes de conduzir o banco de dados aos estados caracterizados pelas conjunções de fatos do predicado 'imposs'. Caso seja possível identificar tais seqüências, há pelo menos uma operação no esquema com pré-condições não suficientes para a preservação das restrições de integridade, situação esta que é comunicada pelo IGP ao ABD.

4.3 A ferramenta de reprojeto

A figura 3 apresenta os componentes da ferramenta de reprojeto e o fluxo de controle, setas simples e duplas tendo o mesmo significado que na ferramenta de projeto e a seta tracejada indicando que as cláusulas do dicionário de alterações são usadas para produzir a saída final, a saber, a nova versão do DD.

Alguns dos componentes da ferramenta de projeto são também usados na fase de reprojeto. Os componentes exclusivos da fase de reprojeto têm as funções seguintes:

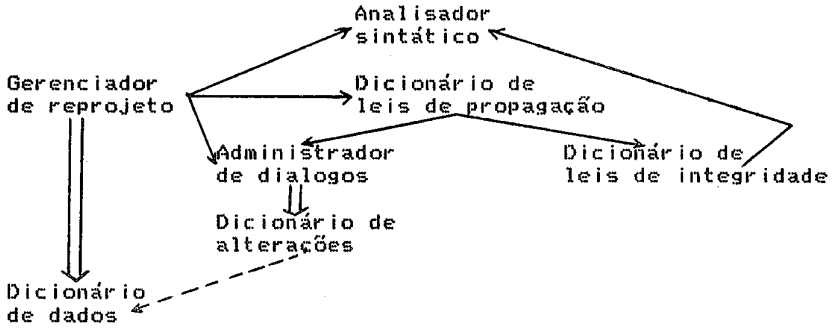
(a) o gerenciador de reprojeto (GRP) coordena a realização de mudanças nas definições de objetos, a introdução de novos objetos e gerencia a propagação da alteração pretendida para os objetos posteriores segundo a ordem de definição;

(b) o dicionário de leis de propagação (DLP) contém uma implementação das leis de propagação da metodologia;

(c) o dicionário de alterações (DA) contém o registro das alte-

rações que tiveram lugar na sessão de reprojeto.

Figura 3: Componentes e fluxo de controle na ferramenta de reprojeto



Da comunicação com o ABD resultam as cláusulas de registro de alterações, que estruturam o DA, as alterações sendo armazenadas na forma de um conjunto de cláusulas definindo um predicado designado "was-applied", que tomará parte então no programa lógico que constitui a ferramenta.

5. Exemplos de utilização do protótipo

São apresentados nesta secção alguns exemplos de uso do protótipo, que correspondem a porções de exemplos completos apresentados no trabalho de [Se], relativos à especificação de um pequeno banco de dados descrevendo as entidades produto e depósito, um relacionamento remessa entre produto e depósito e uma visão entrega sobre o relacionamento remessa.

O primeiro exemplo diz respeito à ativação do gerador de planos. A operação INCLUAPROD do módulo PRODUTO tem um teste para garantir a preservação da restrição de integridade UNICO_N:

```
'UNICO_N (∀p∀n∀m (PROD(p,n) & PROD(p,m) => m=n))'
'INCLUAPROD  ((p,n) :  if ¬∃m(PROD(p,m))
                       then insert (p,n)  into PROD)'
```

Supor agora que o ABD introduz INCLUAPROD sem o teste de garantia, ou seja:

```
'INCLUAPROD ((p,n) : insert (p,n) into PROD)'
```

O IGP construirá, nesta situação, o módulo de aplicação seguinte:

```
'(INCLUAPROD p n) operation-on (pnum nome)'
'(EXCLUAPROD p) operation-on (pnum)'
'(PROD p n) added (INCLUAPROD p n)'
'(PROD p y) deleted (EXCLUAPROD p)'
'(INCLUAPROD p n) conditions ()'
```

"(EXCLUAPROD p) conditions ((PROD p y) (equal p p))"
"imposs ((PROD p n) (PROD p m))"

Da ativação do gerador de planos resultarão os planos (alternativos) seguintes:

"(s0 (INCLUAPROD p n) (INCLUAPROD p m))"
"(s0 (INCLUAPROD p m) (INCLUAPROD p n))"

capazes de produzir uma inconsistência no banco de dados com relação à restrição de integridade UNICO_N. Observe-se que, caso não houvesse o erro, a cláusula "conditions" para INCLUAPROD seria:

"(INCLUAPROD p n) conditions ((not PROD p m))".

O segundo exemplo ilustra o modo de implementação dos requisitos semânticos da metodologia. No atual estágio do protótipo estes são verificados estabelecendo-se uma comunicação com o ABD. Mostra-se abaixo a comunicação levada a efeito para se decidir sobre a introdução ou não de uma cláusula de garantia no esquema:

===")" SEMANTICAL ANALYSIS "("===
* operation INCLUAPROD of module PRODUTO:
* ((p,n) : if -"]"m(PROD(p,m)) then insert (p,n) into PROD)

* constraint UNICO_N of module PRODUTO:
* (\/p\/n\/m(PROD(p,n) & PROD(p,m) "=)" m=n))

* operation INCLUAPROD enforces constraint UNICO_N ?

O terceiro exemplo ilustra a apresentação de mensagens de erro apropriadas quando da violação dos requisitos da metodologia, objetivando nortear o ABD no projeto/reprojeto do esquema. No exemplo o ABD tenta tornar estendido um módulo que já é subordinado no esquema e é advertido pela ferramenta:

* module type --- <type> ?
     Answer is extension

* extends --- <module> ?
     Answer is REMESSA
     Answer is PRODUTO

*** Syntactical error ***
*** E1 requirement violated ***
* module REMESSA subsumes module PRODUTO in the modular schema

Neste fragmento de comunicação com o ABD, as respostas deste aparecem sublinhadas.

6. Conclusões

Como assinalado em [FCT], um aspecto crítico do protótipo é a distribuição da carga de trabalho entre o usuário e a ferramenta especializada. Embora as regras implementadas coordenem com alguma extensão os esforços de projeto e reprojeto do usuário,

muito permanece em aberto para este, não apenas em termos das alternativas válidas (o que é desejável) como também de cometer erros (o que não é desejável), principalmente no que se refere à implementação dos requisitos semânticos. Como se pode observar, obteve-se com este trabalho algum progresso nos aspectos relacionados com a verificação deste tipo de requisito, mediante a integração da ferramenta de projeto com o protótipo de geração de planos.

Na implementação atual do protótipo, efetivou-se a integração da ferramenta de projeto com o protótipo de geração de planos, para verificar/testar a suficiência das pré-condições das operações. Como objetivo de pesquisa futura, para se obter uma ferramenta completa com relação aos objetivos atuais, deve-se executar as etapas de:

(a) integrar a ferramenta de reprojeto com o gerador de planos;
(b) fazer uso do gerador de planos para verificar a correção da tradução de operações (que consta dos substitutivos) de módulos externos [FCT].

Referências Bibliográficas

[BGM] M. Bouzeghoub, G. Gardarin, E. Metais. "Database Design Tools: An Expert System Approach". Proc. of the 11th Int'l. Conf. on Very Large Data Bases, Stockholm, Sweden (1985), 436-447.

[CB] M. A. Casanova, P. A. Bernstein. "A Formal System for Reasoning about Programs Accessing a Relational Data Base". ACM Trans. on Programming Languages and Systems 2:3 (1980), 386-414.

[Ce] S. Ceri. Methodology and tools for database design. North-Holland (1985).

[CFT] M.A. Casanova, A.L. Furtado, L. Tucherman. "A Software Tool for Modular Database Design". Centro Científico IBM Brasil, Relatório Técnico CCB033 (1985).

[CM] K. L. Clark, F. G. McCabe. micro-PROLOG: programming in logic, Prentice-Hall (1984).

[Da] C.J. Date. An Introduction to Database Systems. Addison-Wesley Publishing Company (1976).

[FC] A.L. Furtado, M.A. Casanova. "Updating Relational Views", em Query Processing in Database Systems. W. Kim, D.S. Reiner, D.S. Batory (eds.), Springer-Verlag, (1985), 127-142.

[FCT] A.L. Furtado, M.A. Casanova, L. Tucherman. "A Framework for Design/Redesign Experts". Proc. of the 1st. Int'l Conf. on Expert Database Systems, South-Carolina, Abril (1986), 313-328.

[FM] A.L. Furtado, C.M.D. Moura. "Expert helpers to data-based information systems". Proc. of the First International Workshop on Expert Database systems, (1984), 298-313.

[HS] P. Hammond, M. Sergot. apes: augmented PROLOG for expert systems - reference manual. Logic Based Systems Ltd. (1984).

[Ko] R. Kowalski. Logic for problem solving. North Holand (1979).

[LM] P.C.Lockemann, H.C.Mayr. "Information system design:

techniques and software support" em Information Processing 86. H.J. Kugler (ed.), North-Holland (1986) 617-634.

[Ni] N.J. Nilsson. Principles of artificial intelligence. Springer-Verlag (1982).

[Se] G.J. Sena. Implementação de protótipo de ferramenta para projeto/reprojeto de bancos de dados. Dissertação de Mestrado, PUC/RJ (1987).

[VF] P.A.S. Veloso, A.L. Furtado. "Towards simpler and yet complete formal specifications", em Information Systems: Theoretical and Formal Aspects. A. Sernadas, J. Bubenko, A. Olive (eds.), North-Holland (1985), 175-189.

[Wa] D.H.D. Warren. "WARPLAN: a system for generating plans". Memo 76. University of Edinburgh (1974).