

**GERENCIA DE PROJETO E MECANISMOS GRAFICOS DE  
CONSULTA A BASE DE DADOS NO SISTEMA AMPLO**

Paulo Roberto G. Luzzardi

Carla M. Dal Sasso Freitas

Flávio Rech Wagner

UFRGS - PG em Ciência da Computação

Caixa Postal 1501, 90001 Porto Alegre, RS

**RESUMO** - O trabalho apresenta as funções de controle de acesso e gerência de projeto num ambiente de projeto de sistemas digitais. Ênfase é dada à organização lógica das funções na interface com o usuário e aos mecanismos gráficos de navegação pelos objetos armazenados na base de dados do sistema.

## 1. INTRODUÇÃO

O número e a complexidade de ferramentas de projeto de sistemas digitais, especialmente de circuitos VLSI, aumentaram rapidamente nos últimos anos. Paralelamente, a complexidade dos sistemas digitais projetados cresceu consideravelmente de tal forma que começaram a ser discutidos e trabalhados problemas como consistência entre dados de projeto manipulados por diferentes ferramentas, consistência entre representações de um mesmo sistema digital em diferentes níveis de abstração, uniformidade nas interfaces com o usuário das diferentes ferramentas, gerência da base de dados de projeto, gerência do processo de projeto, etc. Em resposta a estas questões inúmeros ambientes de projeto auxiliado por computador têm sido propostos: ADAM<sup>1</sup>, ULYSSES<sup>2</sup>, FACE<sup>3</sup> e CWS<sup>4</sup>.

Na mesma linha, está em desenvolvimento na UFRGS o sistema AMPLO<sup>5,6</sup>, cujo principal objetivo é oferecer um ambiente aberto, onde estejam integradas ferramentas de projeto tratando diferentes níveis de abstração. AMPLO integra uma família de linguagens de descrição de hardware e facilidades de simulação orientadas para níveis específicos (sistema, transferência entre registradores e portas lógicas). As linguagens são uniformizadas pelos conceitos de *agência* e *sinais de interface* e apresentam versões gráfica e textual equivalentes<sup>7</sup>. Uma agência é um módulo de qualquer complexidade, dotado de um conjunto de sinais de interface, com os quais ele pode se comunicar com

outros módulos, quando da constituição de uma rede (de agências). Um sistema digital (uma agência) pode ser descrito de forma *composta*, na linguagem REDES, como uma rede de outras agências, nada sendo dito a respeito da implementação destas, ou de forma *primitiva*, através das linguagens LAÇO, a nível de sistema, KAPA, a nível de transferência entre registradores e NILO, ao nível de portas lógicas. Agências primitivas ou compostas são consideradas como *definições de tipos*, podendo ser instanciadas na construção de outras agências compostas. Para cada agência podem ser definidas diferentes interfaces, constituindo *alternativas de projeto* da agência. A cada alternativa podem estar associadas diferentes descrições (primitivas ou compostas) correspondendo a *versões da alternativa*. Os objetos de projeto estão, portanto, organizados numa hierarquia agência - alternativas - versões. Assim, sendo uma determinada versão composta, segue a hierarquia com versão - ocorrências de agências. Na construção de uma agência composta podem ser usadas ocorrências de alternativas ou de versões. Uma agência composta construída com ocorrências de versões está pronta para ser simulada. Já uma agência composta por ocorrências de alternativas, necessita de escolha posterior das versões, para simulação.

Uma das questões primordiais, quer seja no desenvolvimento de um ambiente integrado de projeto com ferramentas especialmente construídas, quer seja na integração de ferramentas de vários fornecedores a um ambiente já existente, diz respeito à forma como as facilidades do sistema são colocadas à disposição do usuário e ao monitoramento do processo de projeto <sup>8</sup>. Funções de acesso às várias ferramentas do ambiente, intercâmbio de dados entre diferentes ferramentas, gerência de dados de projeto, particularmente, em relação a diferentes versões de um mesmo sistema digital no mesmo nível de abstração, funções de consulta seletiva (por atributos) aos objetos na base de dados de projeto são alguns dos itens contemplados nas modernas interfaces de ambientes integrados de projeto. AMPLO atende esses requisitos através de uma interface gráfica - LAGO (Linguagem de Acesso Global ao AMPLO) - que provê acesso a todos os recursos de gerência da base de dados, gerência de projeto e ativação das ferramentas do ambiente. Através de um mecanismo de "browsing" podem ser realizadas consultas à base de dados com diversas opções de seleção ("pruning", "poda") na hierarquia de objetos. Tais mecanismos de consulta têm sido largamente utilizados em bases de dados não convencionais <sup>9,10</sup>.

## 2. ORGANIZAÇÃO DAS FUNÇÕES

5  
Através da interface LAGO, AMPLO apresenta funções de administração para controle de acesso aos recursos do sistema e para gerência de projeto e de tarefas específicas dentro de um projeto, bem como funções de projeto propriamente ditas, que correspondem à ativação de ferramentas de edição e simulação. Permanentemente, AMPLO oferece facilidades de consulta à base de dados. A figura 1 resume as facilidades oferecidas por AMPLO. O acesso às funções está vinculado à classificação do usuário dentro do sistema. Um "administrador geral" é responsável pelo cadastro de usuários e de grupos de usuários. Em AMPLO, grupos de usuários são criados para realizar projetos, sob a liderança de um usuário "administrador de grupo". Este é indicado pelo administrador geral quando da criação do grupo e realiza funções de gerência de grupo e das bases de dados associadas ao grupo ("contexto" do grupo <sup>11</sup>). Cabe aqui mencionar que a base de dados do AMPLO é dividida logicamente em níveis: base de dados pública, contextos por projeto, base de dados da tarefa.

A administração de grupo envolve funções de manutenção dos usuários ligados ao grupo, gerência de contextos, realizando o intercâmbio de objetos entre a base de dados pública e os contextos, e gerência de projetos, incluindo a vinculação de um contexto a cada projeto que se inicia.

Usuários "projetistas" têm acesso às funções de ativação de ferramentas (editores de texto, editores gráficos, compiladores e simuladores) e às funções de administração de tarefas (uma tarefa corresponde a uma transação longa, a nível de banco de dados: a criação de uma agência, em várias sessões de edição ou a simulação de uma rede de agências, em várias sessões de simulação). A cada tarefa está vinculada uma base de dados temporária, privativa do projetista, removida após o término da tarefa.

As funções de consulta aos diferentes níveis da base de dados (base de dados pública, contextos, base de dados de tarefa) estão disponíveis quando o usuário inicializa o sistema ou, posteriormente, pela seleção da função de consulta.

## 3. MECANISMOS DE NAVEGAÇÃO

Ao inicializar o sistema ou selecionar a função de consulta a qualquer nível da base de dados, LAGO exhibe todas as agências sob a forma de

retângulos. O usuário pode, então, escolher uma das agências da qual será exibida a hierarquia de alternativas e versões (fig. 2a). Através de mecanismos interativos simples, o usuário tem a possibilidade de:

a) indicando a agência raiz da hierarquia, obter uma nova exibição da árvore sob critérios como nível de definição de alternativas (LAÇO, KAPA e NILO), forma de definição de alternativas (gráfica ou textual), número de ordem de alternativas (absoluto ou por nível de definição);

b) indicando uma determinada alternativa, recuperar a representação (gráfica ou textual) completa da interface, obter a exibição da hierarquia parcial com a alternativa indicada na raiz, sob critérios como linguagem de definição das versões (LAÇO, KAPA ou NILO), forma de definição de versões (gráfica ou textual), número de ordem das versões (absoluto ou por linguagem) ou obter, ainda, o conjunto de agências compostas onde a alternativa é usada como sub-agência;

c) indicando uma versão, recuperar sua representação gráfica ou textual, obter sua árvore de composição no caso de versão composta (REDES, fig. 2b) sob critérios como linguagem de definição das sub-agências, forma de definição das sub-agências ou classe das sub-agências (ocorrências de alternativas ou ocorrências de versões) ou, ainda, obter o conjunto de versões compostas onde a versão é utilizada como sub-agência (fig. 2c).

A navegação pelos objetos da base de dados pode ser feita de forma gráfica, com a representação mostrada na figura 2, ou se o usuário optar explicitamente, de forma textual, como as exibições convencionais de diretórios.

#### 4. COMENTARIOS FINAIS

LAGO está atualmente em fase final de implementação, em ambiente PC. Está sendo implementada em C, fazendo uso do software de apoio desenvolvido para o AMPLO <sup>12</sup>. Seu uso experimental nos próximos meses permitirá uma avaliação mais precisa de sua funcionalidade com vistas a atender os requisitos de integração do sistema.

retângulos. O usuário pode, então, escolher uma das agências da qual será exibida a hierarquia de alternativas e versões (fig. 2a). Através de mecanismos interativos simples, o usuário tem a possibilidade de:

a) indicando a agência raiz da hierarquia, obter uma nova exibição da árvore sob critérios como nível de definição de alternativas (LAÇO, KAPA e NILO), forma de definição de alternativas (gráfica ou textual), número de ordem de alternativas (absoluto ou por nível de definição);

b) indicando uma determinada alternativa, recuperar a representação (gráfica ou textual) completa da interface, obter a exibição da hierarquia parcial com a alternativa indicada na raiz, sob critérios como linguagem de definição das versões (LAÇO, KAPA ou NILO), forma de definição de versões (gráfica ou textual), número de ordem das versões (absoluto ou por linguagem) ou obter, ainda, o conjunto de agências compostas onde a alternativa é usada como sub-agência;

c) indicando uma versão, recuperar sua representação gráfica ou textual, obter sua árvore de composição no caso de versão composta (REDES, fig. 2b) sob critérios como linguagem de definição das sub-agências, forma de definição das sub-agências ou classe das sub-agências (ocorrências de alternativas ou ocorrências de versões) ou, ainda, obter o conjunto de versões compostas onde a versão é utilizada como sub-agência (fig. 2c).

A navegação pelos objetos da base de dados pode ser feita de forma gráfica, com a representação mostrada na figura 2, ou se o usuário optar explicitamente, de forma textual, como as exibições convencionais de diretórios.

#### 4. COMENTARIOS FINAIS

LAGO está atualmente em fase final de implementação, em ambiente PC. Está sendo implementada em C, fazendo uso do software de apoio desenvolvido para o AMPLO <sup>12</sup>. Seu uso experimental nos próximos meses permitirá uma avaliação mais precisa de sua funcionalidade com vistas a atender os requisitos de integração do sistema.

## BIBLIOGRAFIA

1. Granacki, J., Knapp, D. & Parker, A. The ADAM Advanced Design AutoMation System: Overview, Planner, and Natural Language Interface. *Proc. 22nd Design Automation Conference*, 1985.
2. Bushnell, M.L. & Director, S.W. VLSI CAD Tool Integration Using the ULYSSES Environment. *Proc. 23rd Design Automation Conference*, 1986.
3. Smith, W.D. et al. FACE Core Environment: the Model and its Application in CAD/CAE Tool Development. *Proc. 26th Design Automation Conference*, 1989.
4. Gotthell, K. et al. The CADLAB Workstation CWS - an Open, Generic System for Tool Integration. *Proc. IFIP Working Group 10.2 Workshop on Tool Integration and Design Environments*, 1988.
5. Wagner, F.R., Freitas, C.M.D.S. & Golendziner, L.G. The AMPLO System - an Integrated Environment for Digital Systems Design. *Proc. IFIP Working Group 10.2 Workshop on Tool Integration and Design Environments*, 1988.
6. Golendziner, L.G., Wagner, F.R. & Freitas, C.M.D.S. Modeling Digital Systems in an Integrated Design Environment. *Intern. Symp. on Computer Hardware Description Languages and Their Applications*, 1989.
7. Wagner, F.R., Freitas, C.M.D.S. & Golendziner, L.G. Equivalência de Descrições Textuais e Gráficas de Sistemas Digitais num Ambiente de CAD. *XIII Seminário Integrado de Software e Hardware, SBC*, 1986.
8. Di Janni, A. A Monitor for Complex CAD Systems. *Proc. 23rd Design Automation Conference*, 1986.
9. Laenens, F.S. & Vermeir, D. Browsing à la Carte in Object-Oriented Databases. *The Computer Journal*, 32(4):333-340, 1989.
10. Gedye, D. & Katz, R. Browsing the Chip Design Database. *Proc. 25th Design Automation Conference*, 1988.
11. Golendziner, L.G. & Boklis, V. Base de Dados do AMPLO. *Relatório Técnico, em preparação*.

12. Freitas, C., Copstein, B. & Olabarriga, S. Ferramentas para Especificação e Controle da Interface com o Usuário do Sistema AMPLO. *Anais II Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens*, 1989.

*Agradecemos a Lia Golendziner e Vânia Boklis, pela definição e especificação da organização da base de dados e seus mecanismos de acesso, e ao CNPq e a FINEP, pelo apoio financeiro.*

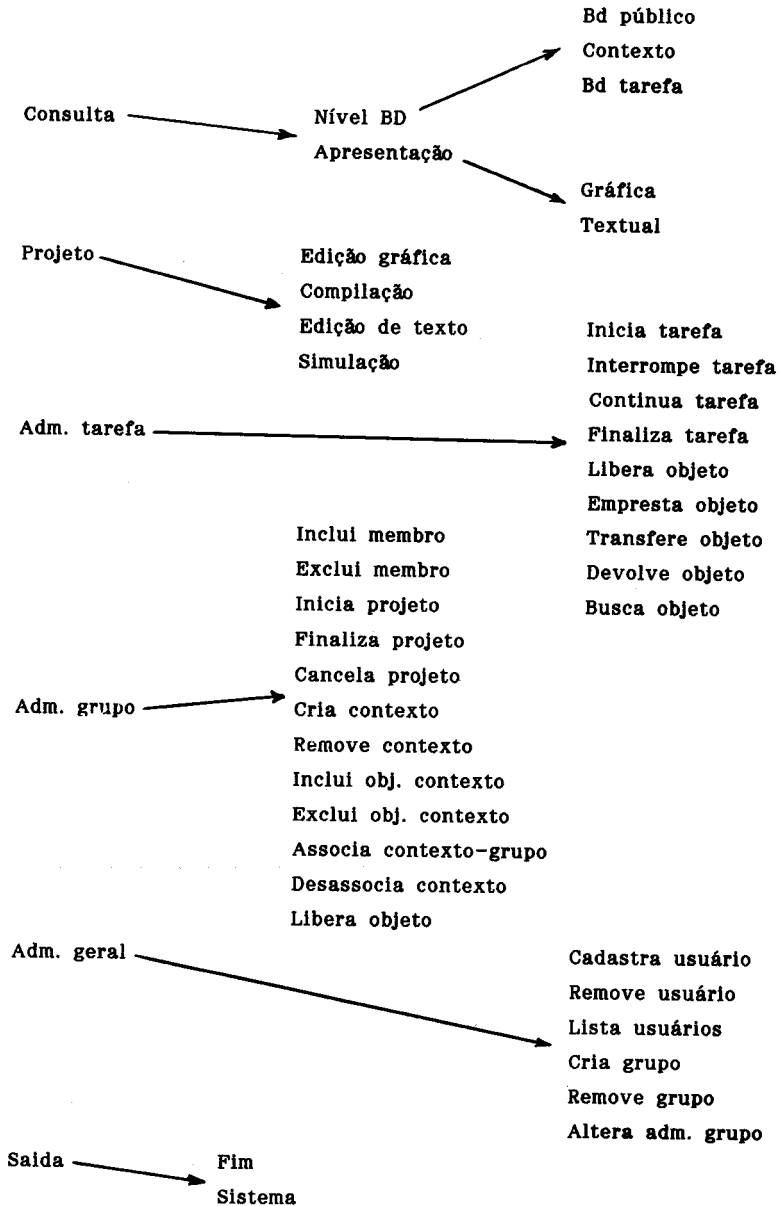


Figura 1 - Resumo das funções oferecidas em LAGO



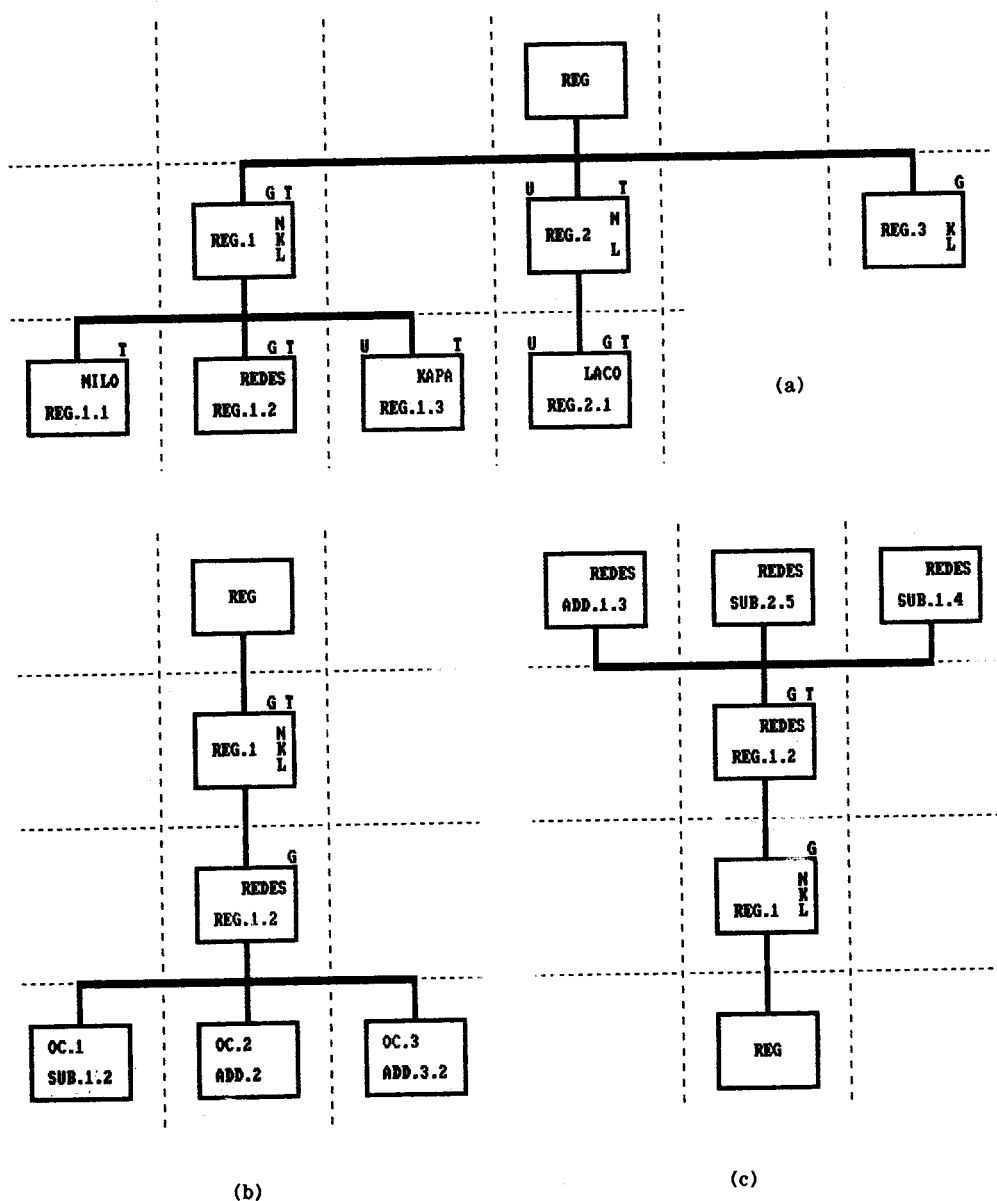


Figura 2 - Representação gráfica de (a) hierarquia de alternativas e versões de agências, (b) hierarquia de versão composta e ocorrências e (c) usos de uma agência em versões compostas.