

MODELAGEM DOS DADOS DE UM  
AMBIENTE DE DESENVOLVIMENTO DE SOFTWARE\*

Javan de Castro Machado, Roberto Tom Price

Curso de Pós Graduação em Ciência da Computação, UFRGS. Caixa  
Postal 1501 - Porto Alegre - RS; Tel: (0512)217714;  
Telex: 051268OCCUFBR. e-mail tomprice@UFRGS.ANSP.BR

RESUMO

Apresenta-se a modelagem dos objetos utilizados pelas ferramentas de um ambiente de desenvolvimento de software. E utilizada a notação diagramática definida para o modelo de dados do Sistema DAMOKLES [DIT 86]. Os objetos apresentam estrutura complexa representando diagramas, programas e textos estruturados resultantes das fases de construção de software.

ABSTRACT

This paper models the objects used by the tools of a software development environment. The diagrammatic notation defined for the DAMOKLES data model [DIT 86] is used. The objects have a complex structure, representing diagrams, programs and structured text resulting from the phases of software development.

1. O AMBIENTE

Os objetos de um ambiente de desenvolvimento de software têm, em sua maioria, estrutura bastante complexa, o que os torna diferentes dos objetos tratados em aplicações convencionais, como os sistemas administrativos. As características não convencionais dos objetos manipulados pelo AMADEUS levaram a modelagem do sistema através de um modelo de dados idealizado para ambientes de desenvolvimento de software: O DODM<sup>2</sup> [DIT 86] do sistema DAMOKLES.

\*1. Adquirido através do projeto financiado pelo CNPQ, FINEP e SBC-UNICAMP.  
<sup>2</sup>Design Object Data Model



O AMADEUS, projeto de um Ambiente de Desenvolvimento de Software (ADS) em construção na UFRGS [FAV 87], [PIM 88], [PRI 88], [MEL 88], [MEL 89], é composto de várias ferramentas integradas que compartilham de uma base de dados comum. As ferramentas interagem entre si através da base de dados que armazena as diversas representações e verbos das especificações e programas produzidos no ambiente. O projeto hoje conta com editores em uso experimental (com instanciação em diversas notações) e ferramentas de transformação de notações. Este trabalho pretende contribuir para a composição do SGBD do ambiente e de outros ambientes.

Todas as ferramentas do AMADEUS geram documentos que, ou fazem parte diretamente do software em desenvolvimento ou são saídas de fases intermediárias do processo de produção do software. Os documentos podem ter especificações gráficas e textuais, diagramas de projeto conceitual do sistema, programas, manuais, massas de testes, etc.

O Editor Dirigido por Sintaxe (EDS) é um editor de textos programável. A sintaxe e semântica da linguagem a ser processada pelo EDS é descrita em LDE (Linguagem de Especificação). A LDE permite a especificação sintática e semântica de uma linguagem, através de uma gramática de atributos estendida por variáveis globais e ações semânticas. O texto em edição é armazenado internamente na forma de uma *Árvore Sintática Abstrata*. Um módulo do sistema cria uma descrição interna (tabelas do reconhecedor) da linguagem a partir da especificação feita em LDE. O EDS e as demais ferramentas que ficam acopladas a ele (depurador, formatador, "browser", compilador, etc) utilizam constantemente as tabelas do reconhecedor na edição, submissão e formatação do texto.

O Editor Diagramático Generalizado (EDG) é uma ferramenta que tem a capacidade de gerar Editores Diagramáticos Especializados. O EDG é dividido em dois módulos principais: O Meta Editor Diagramático (MED) e o Editor Diagramático Específico (EDE). Através do MED são descritos os tipos de nodos, arcos e regras de ligação de uma notação particular. Utilizando o EDE o usuário da técnica interage com o EDG criando e mantendo diagramas. O EDE suporta somente a edição de diagramas de técnicas especificadas no MED. A interface do EDE é uniforme para todas as técnicas suportadas, divergindo somente quanto aos tipos de diagramas que podem ser editados.

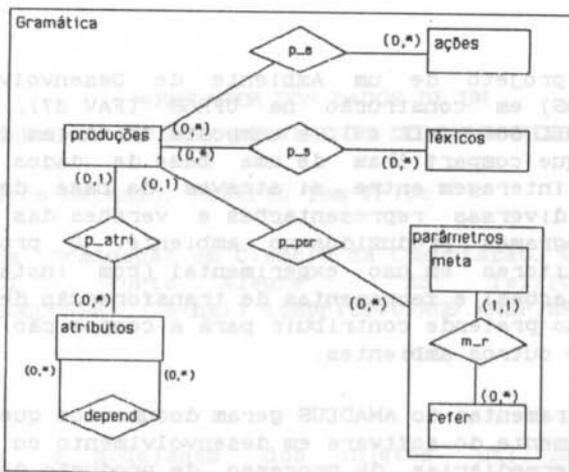


Figura 1: Gramática

O MED é utilizado para a definição dos objetos primitivos da técnica diagramática. Os objetos primitivos são então inseridos num contexto de uma gramática de atributos, especificada em LDE, onde eles são tratados como os elementos léxicos da gramática. A fim de ligar os níveis léxicos e sintático é criado um cabeçalho para todo elemento gráfico [PRI 88]. Com esta finalidade foi estendido o conceito de gramática apresentado acima, incorporando-se a representação de nodos compartilhados para permitir a construção de grafos [PRI 89].

No cabeçalho são especificados atributos herdados e sintetizados. Um elemento gráfico primitivo pode ser tratado como um nodo da gramática de atributos.

A gerência do processo de desenvolvimento consiste do controle de todos os documentos do projeto, as tarefas que os geraram e as pessoas que realizaram tais tarefas. Os gerentes de projeto fazem acesso às informações nos diversos níveis de granulosidade dos documentos. As tarefas, cronogramas e informações sobre os projetistas são alguns dos dados disponíveis para o pessoal de gerência.

## 2. A GRAMÁTICA

Os objetos gramaticais do AMADEUS são mostrados na figura 1. Uma gramática definida em LDE é formada pelos seguintes objetos: Atributos, Ações, Léxicos, Parâmetros, Produções e alguns relacionamentos associando esses objetos.

### 2.1. ATRIBUTOS

A gramática de atributos permite a especificação de atributos associados às produções e a passagem desses atributos entre os nodos das produções. Cada produção pode ter, associadas ao seu cabeçalho, variáveis de importação (atributos herdados), variáveis de exportação (atributos sintetizados) e variáveis de trabalho. Os atributos herdados são recebidos pela produção no momento em que esta é ativada pelo reconhecedor da gramática. Já os valores dos atributos sintetizados retornam quando a produção acaba de ser reconhecida. Variáveis de trabalho são variáveis auxiliares utilizadas durante o reconhecimento da produção. Os atributos são responsáveis por grande parte da verificação semântica estática do texto sendo submetido [Fav 87].

O mecanismo de avaliação dos atributos utiliza um grafo de dependência de atributos no momento da avaliação. Este grafo está representado pelo relacionamento Depend, que liga atributos a atributos determinando quais aqueles que dependem de outros (figura 1). Cada produção pode ou não ter atributos declarados em seu cabeçalho. Esta ligação de pertinência está representada pelo relacionamento P-Atri como mostra a figura 1.

### 2.2. AÇÕES

Ações semânticas são procedimentos ativáveis pelo reconhecedor. As ações são associadas às produções da gramática através do relacionamento P-A (figura 1). Uma ação semântica é como um elemento de uma produção. Quando o reconhecedor encontra uma ação na sucessão de elementos da produção, essa é ativada. As ações são utilizadas para manipulação de tabelas de símbolos, formatação de texto, atribuição de variáveis, avaliação de expressões, etc. Podem ser utilizadas também para verificação da

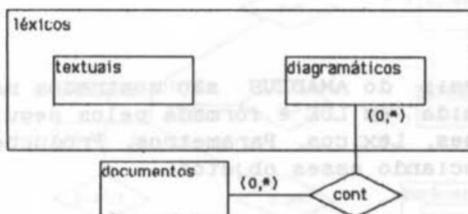


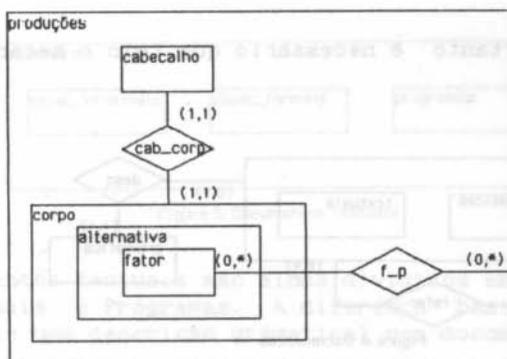
Figura 2: Léxicos

### 2.3. LEXICOS

Os objetos léxicos são divididos em dois grupos: os léxicos textuais e os léxicos diagramáticos (figura 2). Os léxicos textuais são constituídos pelo conjunto básico de caracteres e cadeias de caracteres, formando as unidades léxicas da linguagem. Os léxicos diagramáticos são as unidades léxicas de uma técnica diagramática. Eles contêm todas as características de desenho e formatação dos diagramas, que são definidos através do Meta Editor Diagramático. Os léxicos diagramáticos podem ser nodos, arcos, pontas de arcos e decorações. Eles podem fazer referência a outros documentos inclusive com gramáticas diferentes (relacionamento Cont).

Os nodos são modelados através da composição de objetos gráficos primitivos disponíveis no Meta Editor Diagramático [Melo 89]. Objetos gráficos primitivos podem ser polígonos, polígonos regulares, retângulos, elipses, segmentos de retas, etc. Arcos são objetos gráficos que associam os nodos de um diagrama. O Meta Editor também permite a especificação de tipos de pontas de arcos. Somente dois tipos de pontas podem ser definidos para um tipo de arco. As decorações são figuras formadas pelos objetos primitivos do Meta Editor. Elas ficam associadas a certos tipos de arcos e sua existência depende da existência do arco. As decorações tem como função fornecer semântica às conexões em determinadas técnicas [MEL 89].

Os léxicos formam os símbolos terminais da sintaxe da linguagem ou diagrama. Se a gramática é aplicada a documentos textuais, os léxicos são os textuais. Se a gramática é aplicada a documentos diagramáticos, então os léxicos são diagramáticos.



#### 2.4. PARAMETROS

O objeto **Parâmetros** (figura 1) representa os parâmetros das chamadas das produções da gramática. É formado pelos objetos **Meta**, **Refer** e um relacionamento **M-R** que os associa. O objeto **Meta** armazena as informações sobre os parâmetros formais. Ele descreve as variáveis dos cabeçalhos das produções. O objeto **Refer** armazena as informações sobre os parâmetros reais. Ele contém os valores das variáveis dos cabeçalhos das diversas chamadas recursivas das produções. O relacionamento **M-R** estabelece a ligação entre as variáveis dos cabeçalhos e os seus valores de chamadas.

#### 2.5. PRODUÇÕES

Uma linguagem é descrita em LDE como um conjunto de produções executáveis recursivamente. Uma produção tem o seu **Cabeçalho** onde são declarados os atributos da gramática e um **Corpo** que é composto de várias **Alternativas**. As alternativas são formadas por **Fatores** que podem ser um componente léxico, um literal, uma chamada a uma ação semântica ou uma notação LDE para expressões repetitivas, opcionais, ou parentizadas, que podem ser recursivamente descritas em função de outras produções. A cada fator de uma produção pode estar associado um conjunto de parâmetros. Uma produção é satisfeita se uma de suas alternativas for satisfeita. Uma alternativa é satisfeita se todos os seus fatores forem sequencialmente satisfeitos [Fav 87].

A recursão das produções está evidenciada no relacionamento **F-P** (figura 2) que liga os fatores de uma alternativa de uma produção a outras produções. É óbvio que a recursão não pode

ser infinita, portanto é necessário que todo o mecanismo seja *acíclico*.

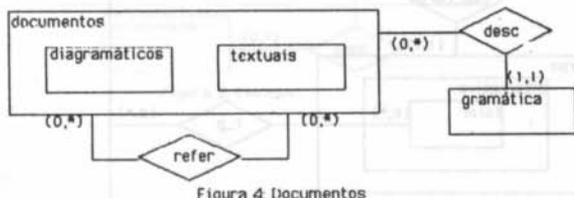


Figura 4 (documentos)

### 3. DOCUMENTOS

O Ambiente suporta diversos tipos de documentos. Cada tipo tem uma gramática associada que descreve a forma geral dos documentos do tipo (figura 4). Os documentos são os vários produtos das tarefas realizadas pelos projetistas ou de tarefas executadas automaticamente pelo ambiente. Documentos de tipos diferentes podem estar associados através do relacionamento Refer. Este funciona como um grafo de controle primitivo entre os documentos. Através do relacionamento Refer pode-se dizer, por exemplo, que uma especificação algébrica se refere a um programa do sistema, ou que um programa implementa determinado processo de um Diagrama de Fluxo de Dados. Os documentos podem ser textuais ou diagramáticos.

#### 3.1. DOCUMENTOS TEXTUAIS

Os documentos textuais são aqueles que utilizam uma linguagem escrita como forma de descrever os documentos do projeto. Eles incluem especificações escritas tais como VDM, português estruturado, especificações algébricas, manuais, programas, dados de teste, etc (figura 5).

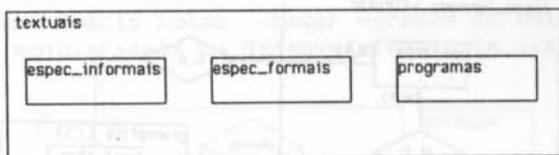


Figura 5: Documentos Textuais

Os documentos textuais são ainda divididos em **Espec-Informais**, **Espec-Formais** e **Programas**. A diferença básica entre eles é a ausência de uma descrição gramatical nos documentos informais.

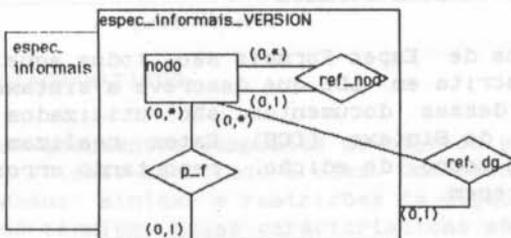


Figura 6: Especificações Informais

### 3.1.1. Especificações Informais

Os documentos informais não possuem gramática livre de contexto descrevendo suas propriedades sintáticas. São editados livremente e são armazenados sob a forma de uma árvore, que contém informações sobre os elementos de composição do documento. Os nodos da árvore contêm a propriedade de fazer referência a outros documentos ou a outras partes do mesmo documento, caracterizando o conceito de *Hipertexto* (figura 6). O editor específico de documentos informais deve ser capaz de reconhecer uma referência e transferir o fluxo do documento para o texto referenciado e manter uma pilha de endereços de retorno. As referências devem ser acíclicas a fim de garantir o fluxo correto entre os documentos.

Os Documentos Informais são utilizados como lembretes que os projetistas podem fazer sobre documentos (especificações, programas, diagramas) que estão editando.

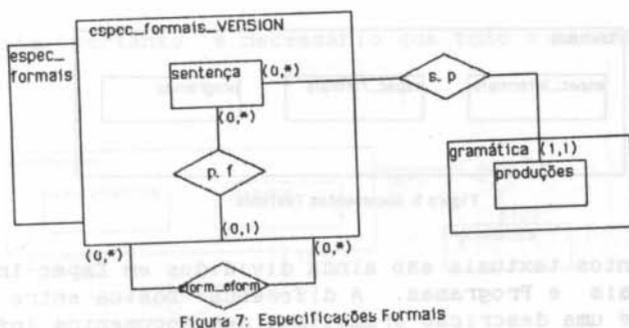


Figura 7: Especificações Formais

### 3.1.2. Especificações Formais

Os Documentos de Espec-Formais são todos aqueles que tem uma gramática escrita em LDE que descreve a sintaxe do documento. Para edição desses documentos são utilizados Editores com Conhecimento de Sintaxe (ECS). Estes realizam a verificação sintática em tempo de edição, reportando erros no momento em que eles aparecem.

Os objetos Espec-Formais podem possuir diversas versões. As versões desses objetos são armazenadas na forma de uma *Árvore Sintática Abstrata* (ASA) como mostra a figura 7. A ASA é atribuída permitindo o uso do mecanismo de gramática de atributos na verificação sintática e semântica do texto.

### 3.1.3. Programas

Os programas são o produto da fase de codificação do ciclo de desenvolvimento. Eles são desenvolvidos numa linguagem de programação que, no ambiente, tem sua gramática especificada em LDE. O *Editor Dirigido por Sintaxe*, que é um ECS, é a ferramenta do ambiente responsável pela edição de programas.

Os programas são compostos por Módulos e estes possuem versões (figura 8). As versões de um Módulo são armazenadas na forma de uma ASA. A ASA representa todas as derivações do reconhecimento de uma sentença da linguagem. Uma sentença pode ser reconstruída através do caminharmento pelas folhas da árvore. A ASA é decorada com atributos que são utilizados pelo mecanismo de gramática de atributos. Os nós da árvore são ligados às produções da gramática através do relacionamento Ref. As

versões de um Módulo podem chamar versões de outros Módulos, permitindo a modularidade no desenvolvimento de sistemas.

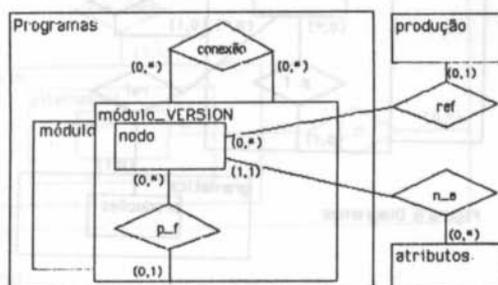


Figura 8: Programas

### 3.2. DOCUMENTOS DIAGRAMATICOS

Os diagramas são documentos compostos de objetos geométricos, podendo ainda ter texto associado a eles. Uma técnica diagramática também possui sintaxe e restrições de ligação entre os vários objetos da técnica. Essas características são especificadas em LDE, onde um mecanismo de gramática de atributos é utilizado. O Meta Editor Diagramático (MED) é utilizado para descrever os objetos diagramáticos da técnica a ser utilizada. O MED possui figuras geométricas que são utilizados como molde para a descrição dos objetos da técnica. O Editor Diagramático Específico utiliza o mecanismo de gramática de atributos para garantir a correta produção de diagramas da técnica. Várias técnicas diagramáticas podem ser especificadas, cada uma delas com uma gramática específica associada.

Os diagramas possuem versões (figura 9). Estas são armazenadas internamente através de uma estrutura de árvore (Árvore Sintática Abstrata). Da mesma forma que os Documentos Formais, os nodos da ASA são associados às produções da gramática. Os léxicos da gramática são os objetos diagramáticos definidos no Meta Editor Diagramático. Um nodo da ASA pode fazer referência a uma versão de um outro diagrama (relacionamento **Explosão** da figura 9), possibilitando a ideia de **Refinamentos** de objetos diagramáticos.

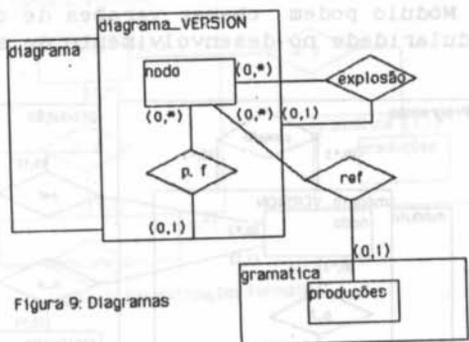


Figura 9: Diagramas

### 3.3. CONFIGURAÇÃO

Uma configuração é um conjunto de verbos e objetos (figura 10). Em geral o pessoal de desenvolvimento trabalha em determinadas partes do projeto e não necessita de todos os objetos de projeto disponíveis numa sessão de trabalho. Uma configuração determina um bloco de documentos que são aglomerados explicitamente pelos projetistas. Pode-se determinar critérios internos para a formação de configuração (p.e. conjunto de rotinas de uma determinada função e os diagramas que as especificaram) e utilizar o mecanismo de gramática de atributos para a determinar incrementalmente os objetos que fazem parte das configurações.

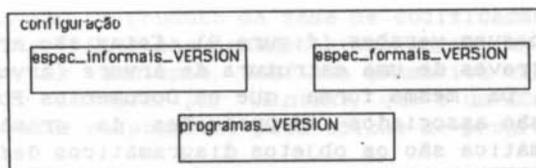


Figura 10: Configuração

#### 4. GERENCIA DO PROJETO

A gerência de projeto consiste basicamente do controle de todos os *documentos* produzidos, *tarefas realizadas* e *pessoal* que trabalha direta ou indiretamente no projeto. A base de dados poderá ser consultada diretamente pelo pessoal de gerência através de uma linguagem de consulta do tipo "Structured Query Language" (SQL). O Sistema Gerenciador de Objetos de Projeto deverá apresentar uma interface amigável com o usuário através de "icons", "mouse", "templates", e outros artificios que tornam fácil a tarefa de consulta à base de dados.

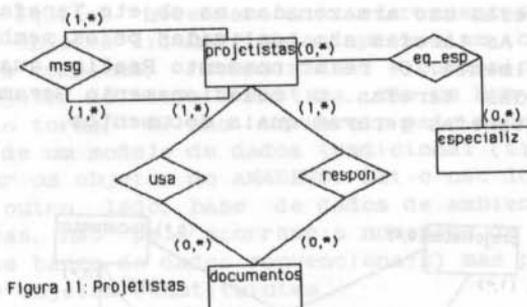


Figura 11: Projetistas

##### 4.1. O PESSOAL DE PROJETO

As pessoas que trabalham no projeto são armazenadas no objeto *Projetistas* (figura 11). Além dos dados gerais de um funcionário, o pessoal apresenta papéis específicos para o projeto em que estão alocados. É importante saber as *especializações* dos membros da equipe para poder alocá-los convenientemente. As especializações ficam armazenadas no objeto *Espec* e se associam com *Projetistas* através do relacionamento *Possue*. Os *Projetistas* *Usam* *documentos* e algumas pessoas são responsáveis (*Respon*) pelos diversos documentos do projeto.

Um dos maiores problemas no aspecto gerencial de projeto é a falha de *comunicação* entre os membros da equipe. AMADEUS deverá prover uma ferramenta de comunicação de mensagens entre as várias pessoas que trabalham no projeto. As mensagens devem ser enviadas pelo *projetistas* para *projetistas* e outros nodos através do auto-relacionamento *Msg* (figura 11).

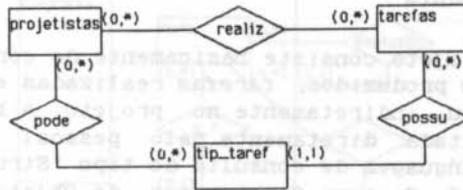


Figura 12: Tarefas

#### 4.2. TAREFAS

As tarefas são os trabalhos designados para as pessoas do projeto. As tarefas são armazenadas no objeto Tarefas como mostra a figura 12. As tarefas são realizadas pelos membros da equipe de desenvolvimento. O relacionamento Realiz guarda aspectos da execução das tarefas. O relacionamento Geram (figura 13) afirma quais tarefas geraram quais documentos.

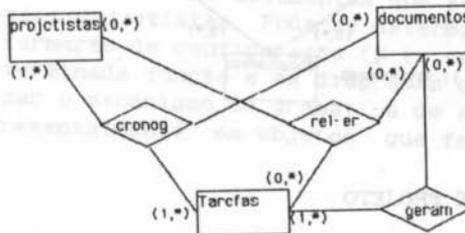


Figura 13: Cronograma e Relatório de Erros

As tarefas são classificadas em tipos. De acordo com o tipo da tarefa ela pode ser designada para um projetista que possui determinadas qualificações. Por exemplo, a tarefa de programação cabe às pessoas que exercem o papel de programador no projeto, a alocação de pessoal é trabalho para o pessoal de gerência, etc. Os tipos de tarefas estão representados pelo objeto Tip\_Taref e se ligam as tarefas pelo relacionamento Possui. O relacionamento Pode determina quais pessoas da equipe estão qualificadas para realizar que tipos de tarefas. Cronograma é um relacionamento entre projetistas, documentos e tarefas. Semanalmente o cronograma é atualizado (figura 13) atribuindo quais tarefas são alocadas para que pessoas da equipe e quais os docu-

mentos envolvidos na tarefa e ainda o prazo para que a tarefa seja realizada.

Os relatórios de erros são produzidos pelos projetistas para reportar a ocorrência de falhas na execução de tarefas ou inconsistência em documentos do projeto. Os relatórios de erros são armazenados através do relacionamento Rel-Er da figura 13.

## 5. CONCLUSAO

O presente trabalho trata da modelagem dos objetos utilizados em um ambiente de desenvolvimento de software. Os objetos modelados são aqueles necessários às ferramentas do ambiente AMADEUS. Os objetos identificados apresentam características de estrutura complexa, recursividade e composição de textos e figuras. Objetos do tipo Gramática são um bom exemplo disso. Esse aspecto torna, se não inviável, pelo menos trabalhosa a utilização de um modelo de dados tradicional (tipo relacional) para modelar os objetos do AMADEUS. Daí o uso do DODM na modelagem. Por outro lado, base de dados de ambientes de projeto são volumosas, não pela ocorrência numerosa de tipos de objetos (como em banco de dados convencionais) mas pela composição complexa dos objetos constituintes.

As ferramentas do ambiente trabalham de forma integrada e compartilham de uma base de dados comum. Dicionário de dados e grafos de controle determinam a interação e ordem de utilização das várias ferramentas do ambiente.

Em função dessa modelagem, está-se trabalhando no momento na definição do Sistema de Gerencia de Base de Dados que atenda as necessidades de ambientes de desenvolvimento de software e em particular a categoria de ambientes dirigidos pela sintaxe como o AMADEUS.

## REFERENCIAS BIBLIOGRAFICAS

- [DIT 86] DITTRICH, K.R. et al. DAMOKLES -- A Database System for software Engineering environments. In: WORKSHOP ON ADVANCED PROGRAMMING ENVIRONMENTS, Trondheim, June 16--18, 1986. Proceedings. Norwegian, Springer Verlag, 1986 p.353-71,

- [FAV 87] FAVERO, E.L. A implementação de um núcleo de um gerador de editores dirigidos por sintaxe em um microcomputador. Porto Alegre, PGCC da UFRGS, 1987.
- [KAR 88] KARLSRUHE, F.Z.I. DAMOKLES - Database Management System for Design Applications - Reference Manual. Mar. 1988.
- [MEL 88] MELO, W.L.M. & PRICE, R.T. Implementação de um Editor de Diagramas Generalizado. In SIMPOSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 2. Canela, out. 27-28, 1988. Anais. Porto Alegre, DI-UFRGS, 1988. p.123-8.
- [PEN 85] PENEDO, M.H. & STUCKLE, E.D. PMDB - A Project Master Database for Software Engineering Environments. In INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 8. London, Aug. 28-30, 1985. Proceedings. Washington, Computer Science, 1985. p.150-7.
- [PIM 88] PIMENTA, M.S. & PRICE, R.T. Um depurador multilinguagem em um ambiente de desenvolvimento. In SIMPOSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 2. Canela, out. 27-28, 1988. Anais. Porto Alegre, DI-UFRGS, 1988. p.86-95.
- [PRI 84] PRICE, R.T. A prototype syntax driven language editor. Brighton, University of Sussex, 1984. (Tese de Doutorado).
- [PRI 88] PRICE, R.T. & FAVERO, E.L. Editores Diagramáticos baseados em Formalismos Gramaticais. In: JORNADA ARGENTINA DE INFORMATICA, 17. Buenos Aires, Set. 26-30, 1988. Anais. Buenos Aires, SADIO, 1988. p.509-27.
- [PRI 89] PRICE, R.T. & FAVERO, E.L. Um editor híbrido (textos e diagramas) orientado por estruturas (do tipo grafo e árvore). A ser publicado em III Simpósio Brasileiro de Engenharia de Software. Recife - PE. out. 1989.