

## Um dicionário de dados para um editor diagramático generalizado

Flávio Roberto Mrack \*,+  
 Mônica Spotorno da Silva \*,+  
 Roberto Tom Price \*\*, -

### Sumário

É descrita a integração de um Dicionário de Dados (DD) ao Editor Diagramático Generalizado (EDG) - um gerador de editores diagramáticos especializados em notações definidas pelo usuário e possíveis de serem abstraídas como grafos.

O dicionário de dados armazena dados gerados pelos diagramas e dados associados pelo usuário aos símbolos diagramáticos. Além disso, o DD permite a amarração de diversas técnicas diagramáticas, efetuando verificações de consistência e possibilitando ao usuário criar metodologias de desenvolvimento de software.

### Abstract

The integration of a Data Dictionary (DD) associated to a Generator of Diagrammatic Editors (EDG) is described. EDG is an editor generator that allows user definition of graph-like diagrammatic notations.

The DD stores data generated by the diagrams and data associated by the user to the diagrammatic symbols. Furthermore, the DD allows the integration of several diagrammatic techniques through the access to the corresponding DD entries enabling consistency checking among techniques, thus in fact creating means to define software development methodologies.



Este trabalho está sendo desenvolvido com o apoio financeiro do CNPq, FINEP, SID-Informática e projeto ETHOS.

\* Mestrando em Ciência da Computação (UFRGS/RS)

\*\* Professor e Pesquisador (UFRGS/RS), Eng, MSc, DPhil

+ Universidade Federal do Rio Grande do Sul, Curso de Pós-Graduação em Ciências da Computação, Caixa Postal 1501, Porto Alegre, RS. Mail: TOMPRICE@SBU.UFRGS.ANRS

## 1 Introdução

São várias as fases do ciclo de desenvolvimento de um sistema. A fim de obter uma melhor qualidade do sistema e permitir uma visualização mais fácil de seus objetivos, são utilizadas técnicas diagramáticas, principalmente nas fases de análise e projeto do sistema. Como exemplo de técnica diagramática, cita-se o diagrama de entidade e relacionamento [CHE76], utilizado na fase de modelagem de dados.

O **Editor Diagramático Generalizado (EDG)** [MEL 88][MEL 89], implementado no Projeto **Ambiente e Metodologia Adaptáveis de Desenvolvimento Unificado de Software - AMADEUS**, é uma ferramenta capaz de gerar editores diagramáticos especializados, fornecendo mecanismos pelos quais diagramas possíveis de serem abstraídos como grafos possam ser descritos. Com base nesta descrição o **EDG** fornece facilidades para a edição de diagramas na técnica especificada.

Visando capturar uma maior quantidade de informações dos elementos que compõem um diagrama, são utilizados Dicionários de Dados (DD). Um DD é uma ferramenta capaz de armazenar e recuperar informações referentes aos elementos de um diagrama, e de definir correspondências (relações) entre estes elementos, podendo avaliar a consistência do diagrama editado.

Durante as etapas do desenvolvimento de uma determinada aplicação, podem ser utilizadas diversas técnicas diagramáticas - por exemplo, numa fase inicial podem ser modelados dados, através de diagramas de entidade e relacionamento, e após pode ser descrito o fluxo destes entre os processos de transformação, através de diagramas de fluxo de dados [DEM76]. Os resultados produzidos pela edição de diagramas de uma técnica diagramática servem de suporte à edição de diagramas na fase seguinte. Com isto, integram-se os diagramas que descrevem uma aplicação, através de uma metodologia de desenvolvimento de software.

O **EDG** suporta a criação de uma metodologia de desenvolvimento de software, armazenando as informações capturadas pelo sistema em um protótipo de dicionário de dados [MRA 89] baseado no modelo relacional. O protótipo permite que cada aplicação defina o modelo conceitual de dicionário adequado para armazenar as informações dos elementos que compõem os diagramas da metodologia.

A fim de permitir que os resultados de uma fase do ciclo de desenvolvimento do sistema dêem suporte à fase seguinte, o DD do **EDG** permite estabelecer correspondências entre objetos de diferentes técnicas diagramáticas. Neste trabalho, objeto significa a instância de um símbolo de uma técnica diagramática. Por exemplo, a ocorrência do processo Enviar Pedido em um diagrama de fluxo de dados é um objeto, e processo é um símbolo da técnica diagramática diagrama de fluxo de dados.

## 2 O Editor Diagramático Generalizado (EDG)

O EDG suporta duas classes de usuários: o usuário especificador ou administrador de CASE (AC) [MAR 86], responsável pela criação e manutenção de novas instâncias de técnicas diagramáticas, e o usuário final, normalmente um analista de sistemas, que edita um diagrama de uma técnica diagramática previamente especificada pelo AC.

A figura 1 mostra o fluxo de funcionamento do EDG. Como pode ser visto, ele é dividido funcionalmente em três módulos principais: o Descritor de Dicionário de Dados (DDD), tratado na seção 3, o Meta Editor Diagramático (MED) e o Editor Diagramático Específico (EDE)

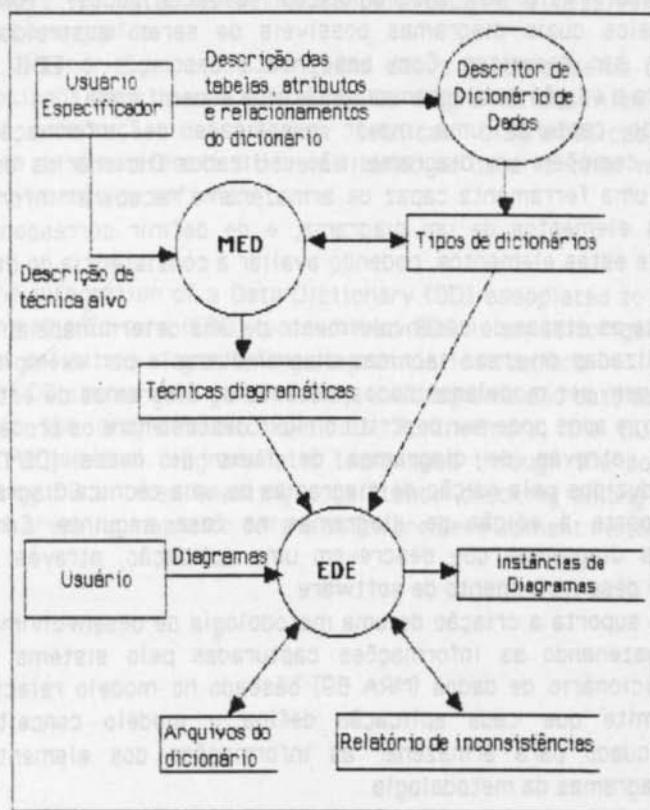


Figure 1 - Diagrama de fluxo de dados do EDG.

### 2.1 O Meta-Editor Diagramático (MED)

Por intermédio do MED são realizadas as especificações dos tipos de diagramas suportados no EDG. Para definir tais técnicas, o MED suporta cinco tipos de entidade: nós, arcos, decorações, pontas e regras de ligações. Tais

tipos de entidade possibilitam ao AC definir as principais características diferenciadoras das técnicas desejadas, e são descritas a seguir.

Os tipos de nodos são modelados através da composição de objetos gráficos primitivos disponíveis no **MED**, sendo estes de oito tipos: polígonos, polígonos regulares, retângulos, retângulos chanfrados, elipses, segmentos de retas, segmentos curvos e textos. Cada objeto criado no **MED** possui atributos de visualização, como cor de fundo, cor e largura da(s) linha(s) de contorno, dimensões, além de atributos de identificação e número possível de conexões mínimas e máximas. Na figura 2, são mostrados três tipos de nodos definidos para a técnica diagrama de fluxo de dados: "Entidade Externa", "Função" e "Arquivo".

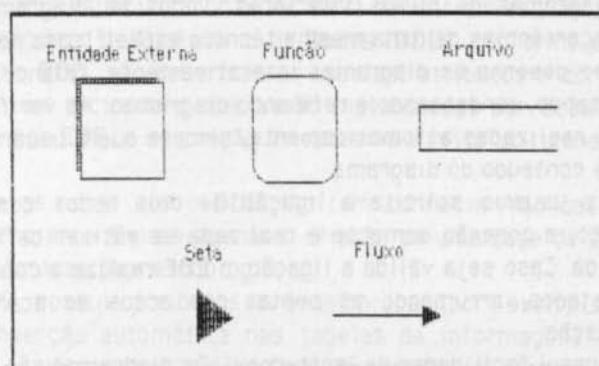


Figura 2 - Nodos, pontas e arcos definidos para a técnica de diagrama de fluxo de dados.

Os arcos são formados no **MED** pela composição de até duas pontas (também definidas no **MED**) com uma linha, que pode ter a sua largura e padrão especificados pelo AC. Os tipos de pontas são desenhados da mesma forma que os tipos de nodos. Na figura 2, pode-se ver a ponta "Seta", utilizada na composição do arco "Fluxo".

A maioria das técnicas possuem restrições de conexão entre nodos, isto é, certos tipos de nodos só podem ser conectados a determinados tipos de nodos e através de tipos específicos de arcos. Alguns nodos ainda possuem portas de ligação, por onde devem ser realizadas as conexões. As regras de ligação são descritas no **MED** através de uma tabela de ligações possíveis, informando-se quais nodos podem ser conectados através de quais arcos.

Decorações são figuras formadas pelos objetos primitivos do **MED**, que podem estar associadas a certos tipos de arcos. Sua função é fornecer sentido semântico às conexões em determinadas técnicas. As decorações não podem existir sem a presença do arco do qual dependem. Os tipos de decorações são criados no **MED** da mesma forma que os tipos de nodos.

## 2.2 O EDE : Editor Diagramático Específico

O usuário do método alvo interage com o **EDG** através do **EDE**, criando e mantendo diagramas. O **EDE** somente permite que sejam manipulados diagramas de métodos especificados no **MED**. A interface do **EDE** é uniforme para todas as técnicas suportadas, divergindo, é claro, somente quanto aos tipos de diagramas que podem ser editados.

A entidade manipulada pelo **EDE** é o documento. Um documento é representado em forma de árvore onde cada nó é um diagrama. Ou seja, um documento pode ter vários níveis de diagramas.

Em cada diagrama o usuário pode reproduzir vários nodos, decorações e arcos de uma técnica previamente especificada no **MED**. Os nodos podem ser refinados em diagramas de níveis inferiores. Todos os diagramas de um documento são ocorrências de uma mesma técnica especificada no **MED**.

O usuário desenha os diagramas interativamente, criando, movendo, excluindo, conectando, condensando e refinando diagramas. As verificações de consistência são realizadas automaticamente, sempre que o usuário realizar uma alteração de conteúdo do diagrama.

Quando o usuário solicita a ligação de dois nodos com um tipo particular de arco, a conexão somente é realizada se estiver definida como uma ligação válida. Caso seja válida a ligação, o **EDE** realiza a conexão entre os objetos desejados, arrumando as pontas dos arcos de acordo com a inclinação da ligação.

O **EDE** possui facilidades de "soft-copy". Os diagramas são impressos como foram desenhados, sendo que os vários níveis de refinamento são mostrados de acordo com o desenho. Caso um diagrama possua mais que uma página de impressão o usuário poderá solicitar a redução deste para somente uma página, ou imprimi-lo em várias páginas.

O **EDE** também dispõe de um editor de textos com todas as facilidades de navegação, procura e troca de caracteres. A edição de textos pode ser realizada dentro do limite do objeto (menor retângulo que contém todos os pontos do objeto) ou livre no diagrama. No primeiro caso, o texto não pode ultrapassar os limites, logo o texto é rolado dentro do objeto quando necessário. No segundo caso, os textos são editados livremente dentro do diagrama. Os textos podem ser estilizados, alinhados e ter os fontes trocados. Aqueles que estiverem associados a nodos, arcos e decorações acompanharão as translações que estes sofrerem.

O **EDE** também possui facilidades de formatação automática de grafos hierárquicos (SIL 89), uma opção acionável pelo usuário através do cardápio.

À medida que o diagrama é editado, o usuário pode solicitar o refinamento dos nodos. O **EDE** cria uma nova janela para o edição do diagrama de refinamento do nodo desejado. O novo diagrama conterá todas as interfaces do nodo pai no diagrama superior. As operações de exclusão de

ligações no nodo pai afetam os diagramas filhos, e também, as alterações na interface do diagrama filho alteram as conexões no diagrama pai. Desta forma a consistência sintática do documento é mantida.

### 3 Definindo um tipo de dicionário no EDG

Nesta seção e na próxima, à medida que são descritas as facilidades oferecidas pelo dicionário de dados, é dado um exemplo de desenvolvimento de uma aplicação envolvendo as técnicas de diagrama entidade/reacionamento e diagrama de fluxo de dados.

O usuário especificador ou AC deve definir um esquema de dicionário no módulo **Descritor de Dicionário de Dados (DDD)**, conforme pode ser visto na figura 1. No **DDD**, são descritas as tabelas, atributos e correspondências que compõem um dicionário, que serão instanciadas por ocasião da criação dos diagramas no **EDE**. A descrição do esquema é feita utilizando-se o modelo relacional.

Deve-se notar que o **DDD** não tem conhecimento das técnicas que compõem o ambiente. Existe apenas a criação de uma base de dados comum à todas as técnicas da metodologia.

O **DD** possui atributos pré-definidos (por exemplo, "Key"), que efetuam a inserção automática nas tabelas de informações capturadas no diagrama. O atributo "Key" identifica um objeto de forma inequívoca no dicionário, correspondendo a uma composição do nome do objeto no diagrama, nome do símbolo do objeto, nome da técnica diagramática e nome do diagrama. Há uma restrição do dicionário que obriga a presença deste atributo em todas as tabelas definidas pelo usuário.

Além dos atributos pré-definidos, o usuário pode definir quaisquer tipos de atributos em uma tabela. Para cada atributo definido pelo usuário, o usuário deve descrever o seu tipo: Char(n), Byte, Integer ou Real. Os atributos definidos pelo usuário podem ser formados apenas por estes tipos básicos.

O dicionário desconhece o significado semântico dos atributos definidos pelo usuário. Pretende-se avaliar a semântica das tabelas, atributos e correspondências associando o **DD** a um mecanismo de gramática de atributos, utilizado para definir sintática e semanticamente diagramas [FAV 89].

A especificação das tabelas e atributos do **DD** é feita no diálogo mostrado na figura 3. Devem ser definidas as tabelas do esquema e os atributos pertencentes à tabela (o atributo "Key" é inserido automaticamente), juntamente com os tipos destes.

O **DD** permite definir qualquer tipo de correspondência binária entre dois objetos, não havendo correspondências pré-definidas. No **DDD**, devem ser especificados os tipos de correspondências que existem entre os símbolos

das técnicas diagramáticas que compõem a metodologia.

**Data Description Language of Metodologia ER-DFD**

| Tables  | Attributes  |
|---|---|
| <div style="border: 1px solid black; padding: 5px;"> <p><b>processo</b></p> <p>deposito</p> <p>atributo</p> <p>entidade</p> </div>  | <div style="border: 1px solid black; padding: 5px;"> <p>key</p> <p>descricao</p> <p><b>autor</b></p> </div>                                       |
| <input checked="" type="radio"/> Char <input style="width: 40px;" type="text" value="32"/><br><input type="radio"/> Integer<br><input type="radio"/> Real<br><input type="radio"/> Byte | <input type="radio"/> Index<br><input type="radio"/> Unique index<br><input checked="" type="radio"/> Not index<br><input type="radio"/> Not null |
| <input checked="" type="radio"/> Create<br><input type="radio"/> Remove   | <input type="radio"/> Table<br><input checked="" type="radio"/> Attribute   |
| <div style="border: 1px solid black; padding: 5px; display: inline-block;">Insert text</div>  |   |
| Enter data: <input style="width: 200px;" type="text" value="frequencia"/>   | <input type="button" value="OK"/>   |

Figura 3 - Diálogo de definição de tabelas e atributos na criação do atributo freqüência de tabela processo. "Freqüência" foi definido como sendo do tipo Char (32). Nota-se que já foram definidas as seguintes tabelas para a metodologia exemplo: processo, depósito, atributo e entidade. Os atributos são definidos sobre a tabela correntemente selecionada, no caso processo.

O DD detecta as possíveis inconsistências dos diagramas através das correspondências. Para cada tipo de correspondência definido, o AC deve informar o sentido de propagação das inconsistências. Por exemplo, no diálogo de definição dos tipos de correspondências, mostrado na figura 4, que define o tipo de correspondência entidade do er corresponde a depósito no dfd para a metodologia exemplo, nota-se pelo item Direction: 1>2 que, havendo a alteração de uma entidade no er que se corresponda com um depósito no dfd, será sinalizado ao usuário do dfd que houve alteração na entidade, afetando a consistência do depósito. Uma alteração do depósito não é sinalizada para a entidade, pois não está habilitado o item 2>1. Este tipo de correspondência é chamada de direcionada, pois somente a alteração de um dos objetos afeta o outro. Caso a alteração em qualquer dos objetos da correspondência sempre afete o outro objeto, a correspondência é dita biunívoca. Para definir-se uma correspondência biunívoca, deve-se selecionar o item Direction como sendo 1>2 e 2>1.

Através da definição de correspondências entre objetos de diferentes técnicas diagramáticas, o DD estabelece uma, relação de

dependência entre as técnicas, e efetua operações de consistência sobre seus diagramas. O modo como o sistema avalia a consistência de um diagrama é descrito na próxima seção.

**Definition of Correspondences of Metodologia ER-DFD**

|             |               |
|-------------|---------------|
| Type 1      | entidade      |
| Method 1    | er            |
| Description | corresponde a |
| Type 2      | deposito      |
| Method 2    | dfd           |

Direction:  1 » 2  
 2 » 1

Enter date:

Figura 4 - Definição de tipos de correspondências na metodologia exemplo.

Após definir o esquema do dicionário para aquela metodologia, o AC deve definir no **MED**, para as técnicas diagramáticas que compõem a metodologia, os subesquemas de um modelo conceitual de DD pré-definido no **DDD**. A necessidade de definir o subesquema visa dar proteção e manter a privacidade da base de dados, pois o usuário final de uma determinada técnica geralmente necessita trabalhar com somente uma parte dos dados e não deve possuir meios de acessar as outras informações, o que poderia ocasionar danos, como acesso a informações restritas, alterações indevidas na base de dados, etc.

Há dois tipos de subesquemas que o AC deve definir: uma visão para cada símbolo de técnica diagramática e a visão dos usuários que farão a edição dos diagramas.

A definição da primeira visão é utilizada para determinar as tabelas e atributos a serem instanciadas por um objeto, durante a edição do diagrama no **EDE**. É feita da seguinte forma, no **MED**: cada símbolo da técnica diagramática seleciona as tabelas de um dicionário previamente definido no **DDD** e escolhe os atributos que deseja utilizar.

A segunda visão é utilizada para efetuar consultas à base de dados, e é específica para cada usuário de uma técnica diagramática. Desta forma, é

permitido que os usuários possam consultar dados de outras técnicas diagramáticas, sem contudo dispor de meios de efetuar alterações. Cabe ao AC determinar quais as tabelas e atributos que o usuário deve acessar, havendo a possibilidade de alterar esta visão. Para garantir a segurança, o AC efetua o cadastramento de senhas para os usuários. O AC é, em princípio, o único a ter acesso a todo o esquema, mas pode ceder este direito a outros usuários, definindo uma visão global a estes.

Um exemplo de definição do primeiro tipo de visão é dado na figura 5, com a definição da visão para a técnica dfd. O símbolo seleccionado é fluxo de dados e é o único símbolo a utilizar a tabela fluxo. Todos os atributos da tabela fluxo (key, origin, destination e descricao fluxo) estão seleccionados por fluxo de dados. Observa-se que a técnica diagrama de fluxo de dados também faz uso das tabelas processo e depósito.

Outra facilidade que o protótipo de DD provê ao AC é a impressão do esquema e dos subsquemas definidos para o dicionário em uma metodologia. A impressão é feita em uma linguagem textual semelhante à linguagem de definição de dados de SQL.

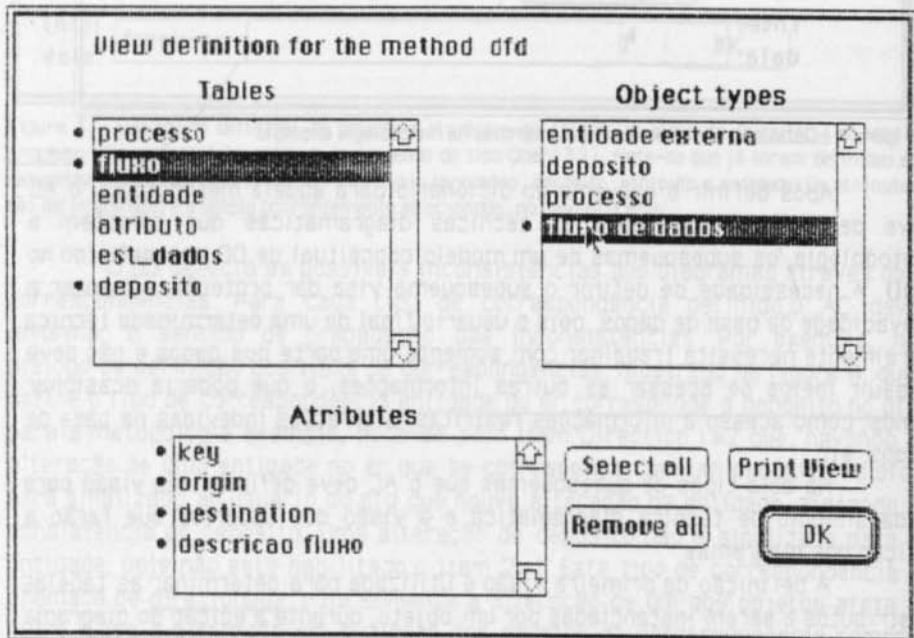


Figura 5 - Definição do subsquema para a técnica diagrama de fluxo de dados.

#### 4 Entrada de dados no dicionário

Foi acrescentado o item Dictionary ao cardápio principal do EDE,

contendo as opções relativas ao dicionário de dados. A figura 6 mostra a opção Dictionary selecionada e as opções disponíveis ao usuário.

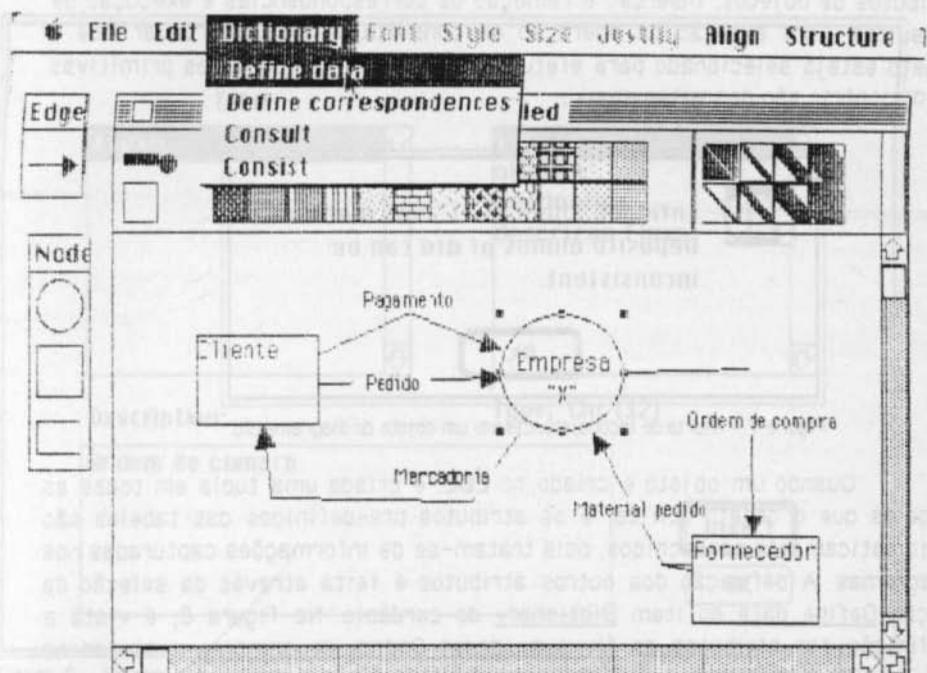


Figura 6 - Opção **D**ictionary do cardápio do EDE. No diagrama da técnica *dfd* note-se que o processo Empresa "X" está selecionado.

O dicionário conta com uma tabela de uso exclusivo do sistema, que não pode ser acessada diretamente pelo usuário. Esta tabela é encarregada de armazenar as inconsistências existentes nos diagramas da metodologia. Na abertura de um diagrama, todas as inconsistências produzidas por objetos de técnicas diagramáticas diferentes da técnica do diagrama atual são apresentadas ao usuário, mostrando qual(is) objetos podem estar inconsistentes. Na figura 7, é dado um exemplo de mensagem de inconsistência. Na edição de um diagrama da técnica *dfd*, é mostrado que em uma alteração ocorrida no diagrama da técnica *er* houve a modificação da entidade aluno, podendo ter afetado a consistência do depósito alunos do *dfd*.

O usuário também pode efetuar uma verificação de consistência entre os objetos da técnica diagramática do diagrama que está editando. Para isto, deve selecionar a opção Consist do item Dictionary do cardápio, mostrada na figura 6. O sistema somente mostrará as inconsistências existentes entre os objetos da técnica diagramática do diagrama corrente.

No EDE, o usuário, à medida que vai editando o diagrama de uma

determinada técnica diagramática, efetua chamadas às operações primitivas do dicionário de dados, que são: inserção e remoção de objetos, alteração de atributos de objetos, inserção e remoção de correspondências e execução de consultas. Com exceção da operação de consulta, o sistema requer que o objeto esteja selecionado para efetuar a operação. As operações primitivas do dicionário são descritas abaixo:

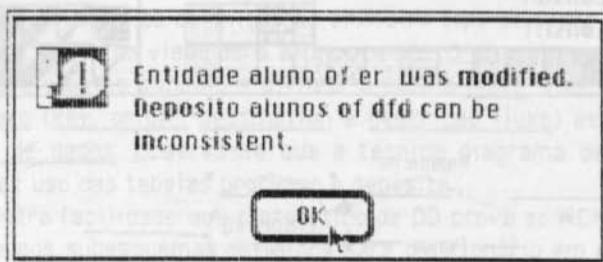


Figura 7 – Alerta de inconsistência em um objeto do diagrama dfd

Quando um objeto é criado no **EDE**, é criada uma tupla em todas as tabelas que o objeto utiliza, e os atributos pré-definidos das tabelas são automaticamente preenchidos, pois tratam-se de informações capturadas nos diagramas. A definição dos outros atributos é feita através da seleção da opção Define data do item Dictionary do cardápio. Na figura 8, é vista a definição dos atributos do fluxo de dados Ordem de compra, mostrado na figura 6.

Na remoção de um objeto, as tuplas são automaticamente removidas das tabelas que o objeto utiliza e, para todas as correspondências nas quais o objeto removido direciona-se a um outro objeto, é colocada uma mensagem de inexistência no arquivo de inconsistências do dicionário, descrito acima. Todas as correspondências relacionadas ao objeto são também removidas.

A alteração de um atributo de objeto é similar à remoção: a tupla da tabela alterada é armazenada novamente no arquivo, e, para todas as correspondências nas quais o objeto alterado direciona-se a um outro objeto, são colocadas mensagens no arquivo de inconsistências. A alteração de um atributo pré-definido pode ser feita somente através de alteração no diagrama, visando proteger a base de dados. Visto que os atributos pré-definidos de um objeto podem estar presentes em várias tabelas, há uma proteção contra alterações acidentais em uma destas tabelas. Uma alteração de um atributo pré-definido propaga-se automaticamente às tabelas em que está definido.

A definição de uma correspondência é feita através da opção Define correspondences no item Dictionary do cardápio, mostrado na figura 6. O sistema exige que ambos os objetos já estejam definidos na base de dados. O sistema mostra os tipos de correspondências possíveis para o objeto

selecionado, e o usuário deve inserir o nome do outro objeto da correspondência.

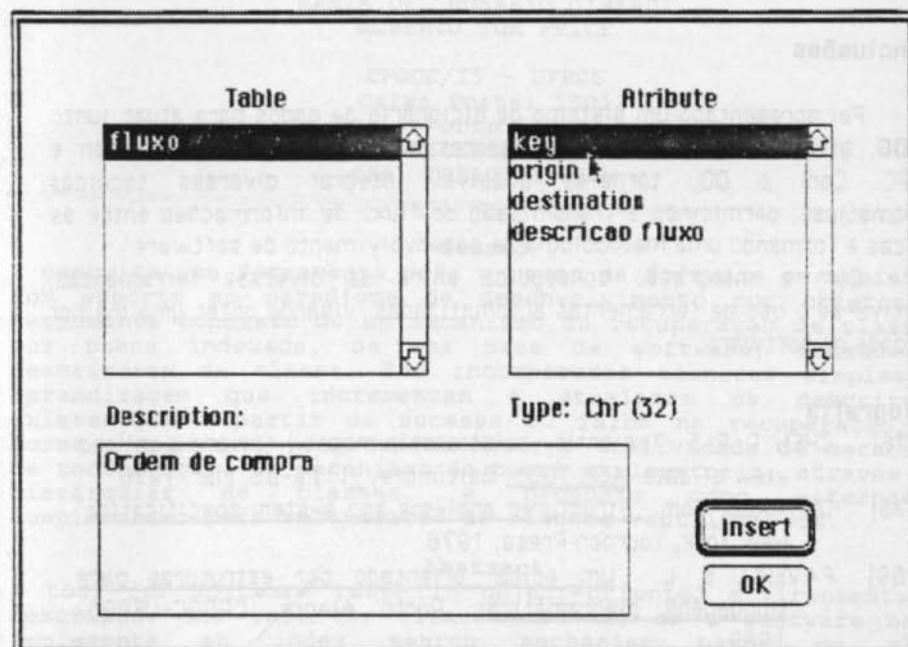


Figura 8 - Entrada de informações no dicionário. É mostrada a definição na tabela fluxo do atributo key, com o valor Ordem de compra. O objeto que está sendo inserido é o fluxo de dados Ordem de compra, mostrado na figura 6. O fluxo de dados pode instanciar os atributos key, origin, destination e descricao fluxo da tabela fluxo, conforme definido na figura 5.

Caso uma correspondência a ser removida esteja direcionada a um outro objeto, é colocada uma mensagem no arquivo de inconsistências do dicionário, sinalizando a remoção da correspondência.

Para acessar rapidamente os dados de um objeto, o sistema conta com um índice para o atributo "Key", implementado sob a forma de árvore-B.

Atualmente, está sendo implementado um mecanismo de consulta que permite a entrada de expressões diretamente em álgebra relacional, a fim de permitir uma maior capacidade de expressão. A entrada da expressão está sendo feita através de uma janela que permite a edição da consulta desejada em uma forma textual.

A partir de uma expressão em álgebra relacional, é gerada uma árvore da expressão. O sistema faz uma comparação entre a visão do usuário e as tabelas requeridas para consulta e, caso o usuário tente acessar tabelas não disponíveis para a sua visão, o sistema indefere a execução da consulta. Caso a consulta seja deferida, o sistema busca as tabelas necessários do disco, e passa a trabalhar na memória principal, gerando tabelas

intermediárias, até obter o resultado da expressão, que são apresentadas em uma janela.

## 5 Conclusões

Foi apresentado um sistema de dicionário de dados para atuar junto ao EDG, atualmente disponível em micros compatíveis com Macintosh e IBM-PC. Com o DD, torna-se possível integrar diversas técnicas diagramáticas, permitindo a transmissão do fluxo de informações entre as técnicas e formando uma metodologia de desenvolvimento de software.

Com a integração conseguida entre as diversas ferramentas, incentiva-se o uso de ferramentas automatizadas, visando obter uma melhor qualidade de software.

## Bibliografia

- [CHE 76] CHEN, P. P. S. The entity-relationship model - toward a unified view of data. ACM TODS, Baltimore, 1(1)9-36, Mar 1976.
- [DEM 76] DEMARCO, Tom. Structured analysis and system specification. New York, Yourdon Press, 1978.
- [FAV 89] FAVERO, E. L. Um editor orientado por estruturas para linguagens diagramáticas. Porto Alegre, CPGCC-UFRGS, 1989.
- [MAR 88] MARTIN, F.C. Second generation Case tools: a challenge to vendors. IEEE Software, Los Alamitos, 5(2)46-9, Mar 1988.
- [MEL 88] MELO, W.L.M. & PRICE, R.T. Implementação de um editor de diagramas generalizado. In: II SBES - SIMPOSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, Canela, Out.27-8, 1988.
- [MEL 89] MELO, W.L.M. A proposta de um editor diagramático generalizado. Porto Alegre, CPGCC - UFRGS, 1989.
- [MRA 89] MRACK, F.R. Protótipo de um dicionário de dados para um editor diagramático generalizado. Porto Alegre, CIC/UFRGS, 1989. (Trabalho de diplomação).
- [SIL 89] SILVA, M.S. Um formatador de diagramas. Porto Alegre, CIC/UFRGS, 1989. (Trabalho de diplomação).