

# Implementação de um Editor de Diagramas Generalizado

Walcélio Louzada Martins Melo \*,+

Roberto Tom Price \*\*,+

## Sumário

As principais características de um Editor Diagramático Generalizado (EDG) são apresentadas. São discutidas as limitações das atuais ferramentas de CASE (Computer-Aided Software Engineering), e sugeridas as facilidades que devem estar presentes na próxima geração de ferramentas. As funções que estão sendo implementadas no EDG são mostradas através de um exemplo de criação de um editor para diagramas de fluxo de dados.

## Abstract

The main features of a Generalized Diagramatic Editor (EDG) are presented. The limitations of currently available CASE (Computer-Aided Software Engineering) tools are discussed and the facilities that must be provided by the next generation of tools are suggested. The functions that are being implemented with EDG are presented. Finally the use of EDG is shown through an example of the creation of an editor for data flow diagrams.

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

\* Tecnólogo em Processamento de Dados (UnB), Analista de Sistemas (SERPRO), mestrando em Ciências da Computação (UFRGS/RS)

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

+ Universidade Federal do Rio Grande do Sul, Curso de Pós-Graduação em Ciências da Computação, Caixa Postal 1501, Porto Alegre, RS

## 1- Introdução

O custo de desenvolvimento de software continua alto e crescente. Alguns estudos [BOE 87] mostram que a taxa de crescimento do custo de desenvolvimento é de 12% ao ano, e que um aumento de 20% na produtividade pode resultar, a nível mundial, em uma economia de US\$ 90 bilhões em 1995.

Porter [POR 85] coloca que mais de 50% do tempo do ciclo de vida de sistemas é gasto realizando ajustes em tarefas concluídas, seja consertando erros de definição ou entendimento da especificação. Logo, a maior oportunidade de aumentar a produtividade no desenvolvimento de software é justamente atacando estes problemas, através do uso de ferramentas e/ou técnicas que possibilitem um melhor acompanhamento da produção de software, evitando que estes erros ocorram. [BOE 87]

Ferramentas de CASE (Computer-Aided Software Engineering) têm sido extensivamente propostas, desenvolvidas e pesquisadas. CASE é uma tecnologia que utiliza ferramentas automatizadas de forma integrada, permitindo que o uso de métodos formais de desenvolvimento tornem-se práticos e econômicos. Através de CASE os analistas e programadores podem obter uma melhor visualização dos requisitos, especificações e projetos de sistemas, além de um acompanhamento automático nos refinamentos sofridos desde a fase de requisitos até o projeto final, permitindo que a "completude" e a consistência sejam verificadas automaticamente [BOE 87] [CHI 88a] [CHI 88b]. Alguns estudos mostraram que o uso de CASE proporciona um ganho de 30-40% nas fases de análise e projeto de sistemas, aumentando também a produtividade nas outras fases. [CHI 88b]

Apesar do aumento da produtividade que ferramentas de CASE introduzem, elas apresentam várias limitações que reduzem esta vantagem, como, por exemplo [MAR 88] [MEL 88a]: a) as ferramentas são confeccionadas para suportar um conjunto pré-definido de técnicas diagramáticas; b) a maioria não possui facilidades para desenho automático; c) a primeira geração de CASEs [MAR 88] enfatiza somente a interação com o usuário e, em sua maioria, são ineficientes na produção de documentação impressa e, d) os CASEs da primeira geração são difíceis de integrar com outros ambientes, pois são construídos de maneira verticalizada.

Estes problemas levam a consideração uma nova geração de CASEs que resolvam estes problemas, proporcionando acréscimos de produtividade. É opinião dos autores que uma maneira adequada de abordar tais problemas é através de um ambiente gerador de CASEs, ou seja, um Meta-CASE.

## 2- Características fundamentais de um Meta-CASE

A principal função de um Meta-CASE, ou "metasystem" [SOR 88], é gerar automaticamente a maior parte de um CASE especializado em uma técnica ou método

particular.

Um meta-CASE caracteriza-se por possuir um administrador de CASE (AC) [MAR 88] responsável pela criação e manutenção de novas instâncias das ferramentas de CASE no ambiente. É através dele que são definidas as convenções e regras de integridade dos tipos de técnicas diagramáticas que o ambiente suporta, como também as verificações de consistência e integridade que devem ser realizadas nos refinamentos e transformações que um diagrama pode sofrer. Um meta-CASE também deve possuir facilidades de "soft-copy" [MAR 88] e formatação automática opcionalmente ativável pelo usuário.

### 3- O projeto EDG

Verificando-se as técnicas de desenvolvimento de software nota-se que a maioria delas utiliza notações diagramáticas, que podem ser abstraídas como grafos. A diferença entre os grafos destas técnicas é encontrada na forma gráfica dos tipos de nodos e arcos, e nas regras de integridade particulares de cada técnica.

É mais conveniente construir um ambiente que suporte várias técnicas diagramáticas, ao invés de construir um CASE para cada grupo restrito. Pensando-se assim teve início em fevereiro de 1988 a construção de um editor diagramático generalizado, denominado EDG, com a função de gerar editores diagramáticos especializados, onde as restrições de integridade e os tipos de nodos e arcos podem ser especificados [MEL 88b].

### 4- A ARQUITETURA DO EDG

O EDG subdivide-se em dois módulos principais (figura 1):

- O Meta Editor Diagramático (MED), utilizado pelo usuário especificador ou administrador de CASEs (AC) [MAR 88] para descrever os tipos de nodos, decorações, arcos (pontas e segmentos) e restrições de integridade utilizadas pela metodologia alvo, para a qual será gerado o EDE.

- O Editor Diagramático Especializado (EDE), encarregado de gerenciar a manipulação de diagramas das metodologias especificadas pelo MED. É utilizado pelo usuário da metodologia alvo.

### 5- O Meta Editor Diagramático

É através do MED que o AD especifica os tipos de diagramas que podem ser editados pela metodologia alvo. Ele subdivide-se em três módulos funcionais: o editor de tipos de diagrama, o descritor das restrições de integridade e o descritor de cardápios.

Através do editor de tipos de diagrama o usuário especificador pode construir graficamente os tipos de arcos, decorações e nodos da metodologia alvo. Este editor

possui uma interface dirigida por cardápios, e utiliza o "mouse" como dispositivo de apontamento e desenho. Nas figuras 2,3 e 4 é mostrada a interface juntamente com um exemplo no qual são definidos os tipos de nodos, arcos e pontas permitidos para um editor especializado em diagramas de fluxo de dados.

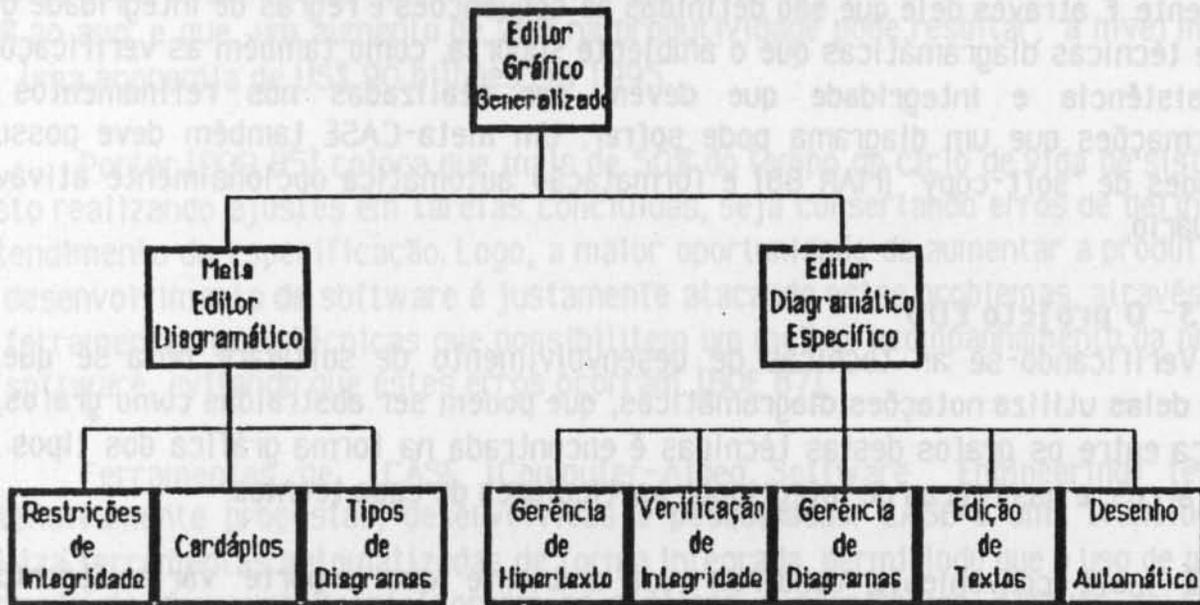


Figura 1 - Estrutura modular do EDG.

No estágio atual do projeto as verificações de integridade estão sendo construídas como programas PROLOG, sendo que outros mecanismos, em especial gramáticas de atributos, estão sendo investigados [FAV 88]. No caso específico das restrições de ligação entre os nodos, é construída a tabela de ligações possíveis para a metodologia alvo. A tabela contém os tipos de nodos que podem ser conectados através de quais tipos de arcos, por quais portas de ligação. O EDG utiliza esta tabela toda vez que é solicitada uma operação de ligação entre dois nodos, impedindo a ligação caso esta não seja permitida [MEL 88b].

Através do descritor de cardápios o AC especifica as funções que serão somadas às já disponíveis no EGE. O AC deve especificar as funções juntamente com o nome das rotinas que as implementam.

## 6- Editor Diagramático Específico

Através do EDE são editados os diagramas da metodologia alvo. Um diagrama pode possuir várias páginas. Podem ser realizadas quaisquer transfêrencias de informações entre as páginas, como também entre diagramas.

O usuário desenha os diagramas interativamente, instanciando, movendo, excluindo, conectando, condensando e refinando diagramas. As verificações de consistência são realizadas automaticamente. Em uma extensão a ser codificada, o usuário poderá solicitar que o EDE realize a arrumação automática das ligações no diagrama, minimizando o número de cruzamentos entre as ligações. Para tal será

utilizada uma extensão do algoritmo proposto por [BAT 86]. Após a arrumação, o sistema voltará a trabalhar no modo Interativo.

As operações de condensação e refinamento dos nodos são gerenciadas pelo EGE, para que a consistência sintática e semântica dos diagramas seja mantida. A operação de refinamento mantém as ligações do nível superior, ou seja, todas as ligações com o nodo pai permanecem após o refinamento, sendo que o usuário da metodologia deve completá-las com os nodos filhos, criados no nível do refinamento.

Através das facilidades de "soft-copy" [MAR 88] o EGE possibilita que os diagramas sejam impressos como foram desenhados.

Encontra-se em fase de especificação um gerente de hipertexto integrado ao EDE. Hipertexto é uma maneira prática e eficiente de documentar várias porções de um sistema/programa sem as restrições dos editores de textos lineares [CON 87]. A integração de hipertexto com CASE facilita a documentação permitindo que as decisões de projeto sejam ligadas com a documentação do programa e com o código fonte do mesmo [BIG 88]. Um diagrama poderá possuir um pedaço (nodo) [BIG 88] [CON 87] do hipertexto associado a cada nível do diagrama. Desta forma, o EDE fará a gerência do hipertexto como uma árvore, onde as ligações ("links") são os refinamentos dos diagramas. Um diagrama pai pode ter vários diagramas filhos, decorrentes dos refinamentos dos nodos no diagrama pai. Assim os diagramas podem ser detalhados até o código fonte do programa que implementará o problema. Com isto, toda a documentação textual vinculada às decisões até a solução podem ser mantidas através do hipertexto associado ao diagrama.

O EDE dispõe de um editor de texto com as facilidades de navegação, procura e troca de "strings". A edição de texto pode ser realizada dentro do envelope dos objetos (menor retângulo que contém todos pontos do objeto) ou livre no diagrama. No primeiro caso o texto não pode ultrapassar os limites do envelope; assim o texto é rolado dentro dele, permitindo que sejam editados textos de até 32K. 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 acompanham as translações que estes sofrerem.

As figuras 5,6 e 7 mostram um exemplo de edição de um diagrama de fluxo de dados, utilizando a definição feita no MED, mostrada nas figuras 2,3 e 4.

## 7- Conclusões

O EDG se configura como um ambiente de suporte para construção de CASEs, isto é, possui todas as características de um meta-CASE. Através do EDG, vários editores específicos podem ser gerados e manipulados de forma homogênea e integrada. Como a interface do EDG é normalizada para todos os editores gerados, a navegação entre eles é realizada sem necessidade de ajustamento por parte do usuário.

Ferramentas de análise e transformação de diagramas podem ser anexadas facilmente ao EDG, pois a estrutura dos dados é armazenada de forma padronizada.

Para que o EDG se constitua um ambiente mais poderoso de auxílio aos desenvolvedores de software é necessário que seja construída uma base de software, onde possam ser armazenados os diagramas juntamente com outros componentes de software. Desta forma, a comunicação entre o EDG e outras ferramentas seria facilitada e incentivada.

Estudos estão sendo feitos no sentido de integrar o EDG com o EDS (Editor Dirigido pela Sintaxe), desenvolvido na UFRGS, possibilitando que textos submetidos sejam analisados sintática e semanticamente.

### 8- Bibliografia

- [BAT 86] BATINI, C.; NARDELLI, E.; TAMASSIA, R. A layout algorithm for data flow diagrams. Trans. Soft. Engineering, SE 12 (4):538-46, Apr. 1986.
- [BOE 87] BOEHM, B. Improving software productivity. Computer, 20 (9):43-57 Sep., 1987.
- [CHI 88a] CHIKOFFSKY, E.J. Software technology people can really use. IEEE Software, Los Alamitos, 5 (2):8-10, Mar. 1988.
- [CHI 88b] CHIKOFFSKY, E.J. & RUBENTEIN, B.L. CASE: reliability engineering for information systems. IEEE Software, Los Alamitos, 5 (2):8-10, Mar. 1988.
- [FAV 88] FAVERO, E.L.; PRICE, R.T. Editores diagramáticos baseados em formalismos gramaticais. (Comunicação Interna). Ago. 1988.
- [MAR 88] MARTIN, F. C. Second-generation CASE tools: a challenge to vendors. IEEE Software, Los Alamitos, 5 (2):46-49, Mar. 1988.
- [MEL 88a] MELO, W.L.M. & PRICE, R.T. Considerações sobre um editor gráfico adaptável a notações diagramáticas. Relatório Interno, CPGCC/UFRGS, fev. 1988.
- [MEL 88b] MELO, W.L.M. & PRICE, R.T. O editor gráfico generalizado. In: CONGRESSO da SOCIEDADE BRASILEIRA de COMPUTAÇÃO, 8., Rio de Janeiro, Jul. 17-22, 1988. Anais. Rio de Janeiro, SBC, 1988. p. 56-66.
- [POR 85] PORTER, M.E., Competitive advantage, Free Press, New York, 1985.
- [SOR 88] SORENSON, P.G.; TREMBLAY, J.P.; MCALLISTER, A.J. The metaview for many specification environments. IEEE Software, Los Alamitos, 5(2):30-38, Mar. 1988.

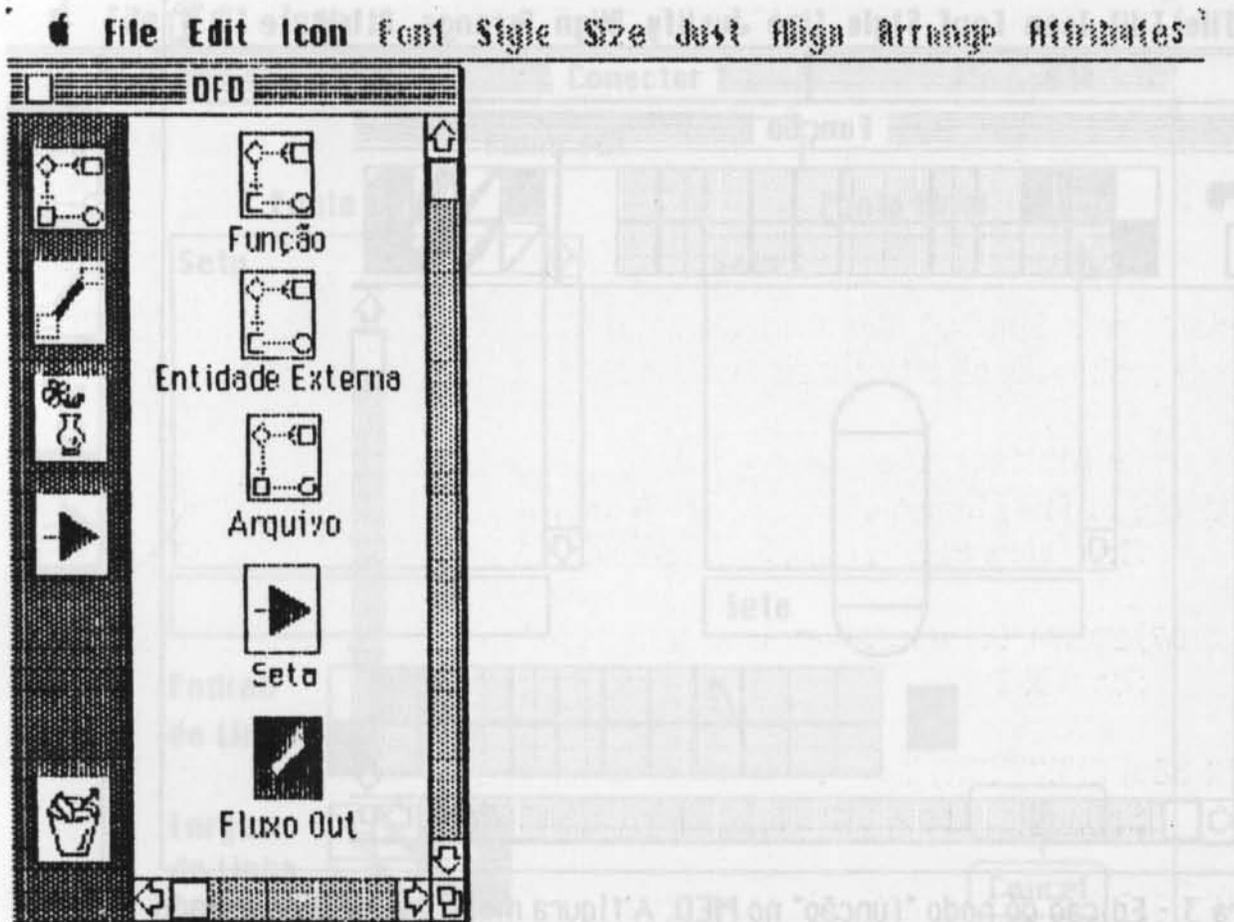


Figura 2- A figura mostra a tela inicial do EDG. Neste exemplo estão sendo construídos os tipos de nodos, arcos e pontas para um editor especializado em diagramas de fluxo de dados (DFD). A tela divide-se em duas partes: a paleta dos ícones disponíveis e a área de catálogo. Existe um ícone para cada tipo de elemento suportado pelo EDG (nodo, arco, ponta e decoração), sendo que o último ícone representa uma lixeira, para exclusão. A área de catálogo mostra os elementos que compõem o EDE para DFD. O MED abriu uma nova janela para cada elemento que o usuário deseja editar.

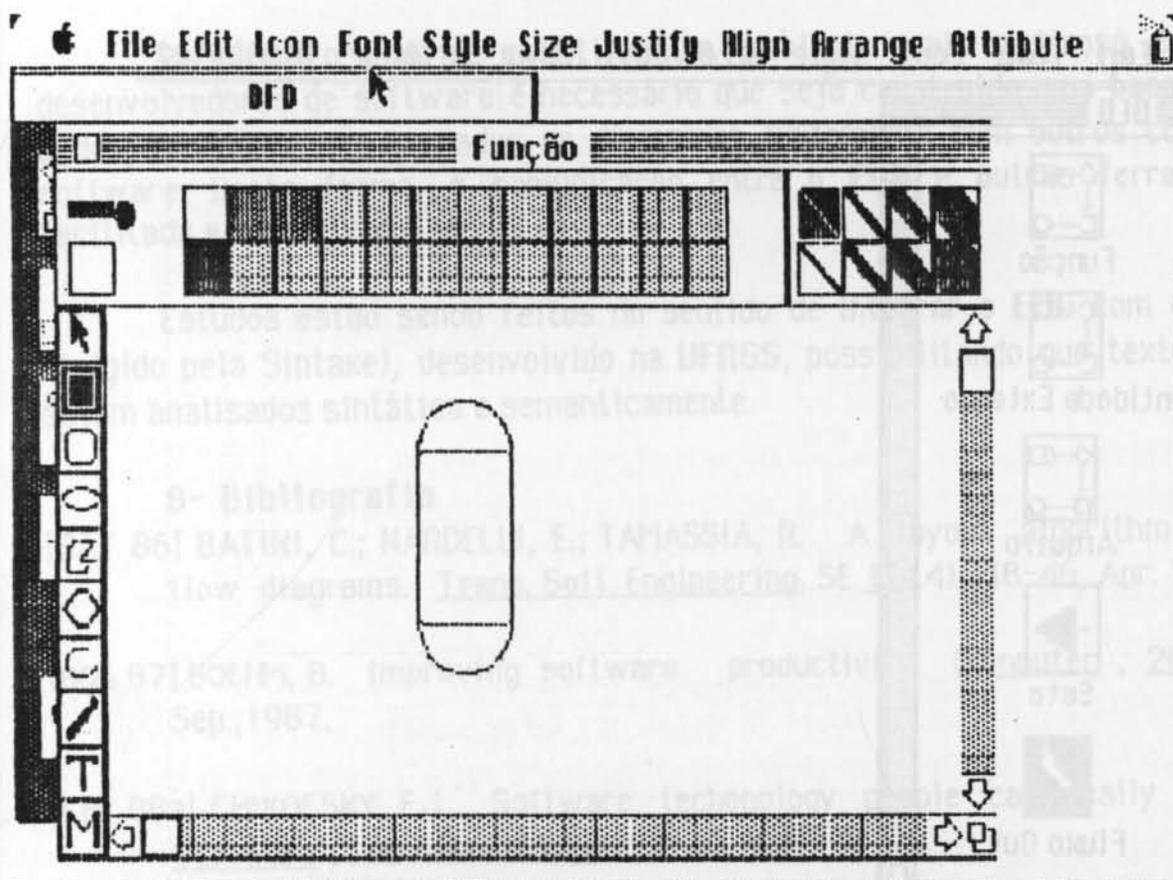


Figura 3 - Edição do nodo "função" no MED. A figura mostra o MED no estado de edição de nodos. Cada nodo é editado em um janela particular. A janela é dividida em quatro áreas: área de padrões de linha e preenchimento disponíveis; área de larguras de linhas; área dos objetos primitivos; e área de desenho. São permitidos 24 padrões diferentes. O usuário dispõe de 8 distintas larguras de linha. Os nodos podem ser compostos por 9 objetos primitivos: retângulos, retângulos com cantos arredondados, elipses, polígonos, polígonos regulares, arcos de circunferências, segmentos de retas, textos e meta-textos. Os textos são editados com auxílio de um editor de textos que possui as facilidades de estilização e navegação. O MED possui 12 funções de alinhamentos de objetos, permite, também, que os objetos sejam rotacionados livremente, além de outras facilidades. As janelas de edição de decorações e pontas são semelhantes a janela de edição de nodos. Não é permitido compor a ponta com texto, meta-texto, retângulo, retângulo arredondado em decorrência dos problemas com rotações que a ponta poderá sofrer no EGE.

Mar, 1986.

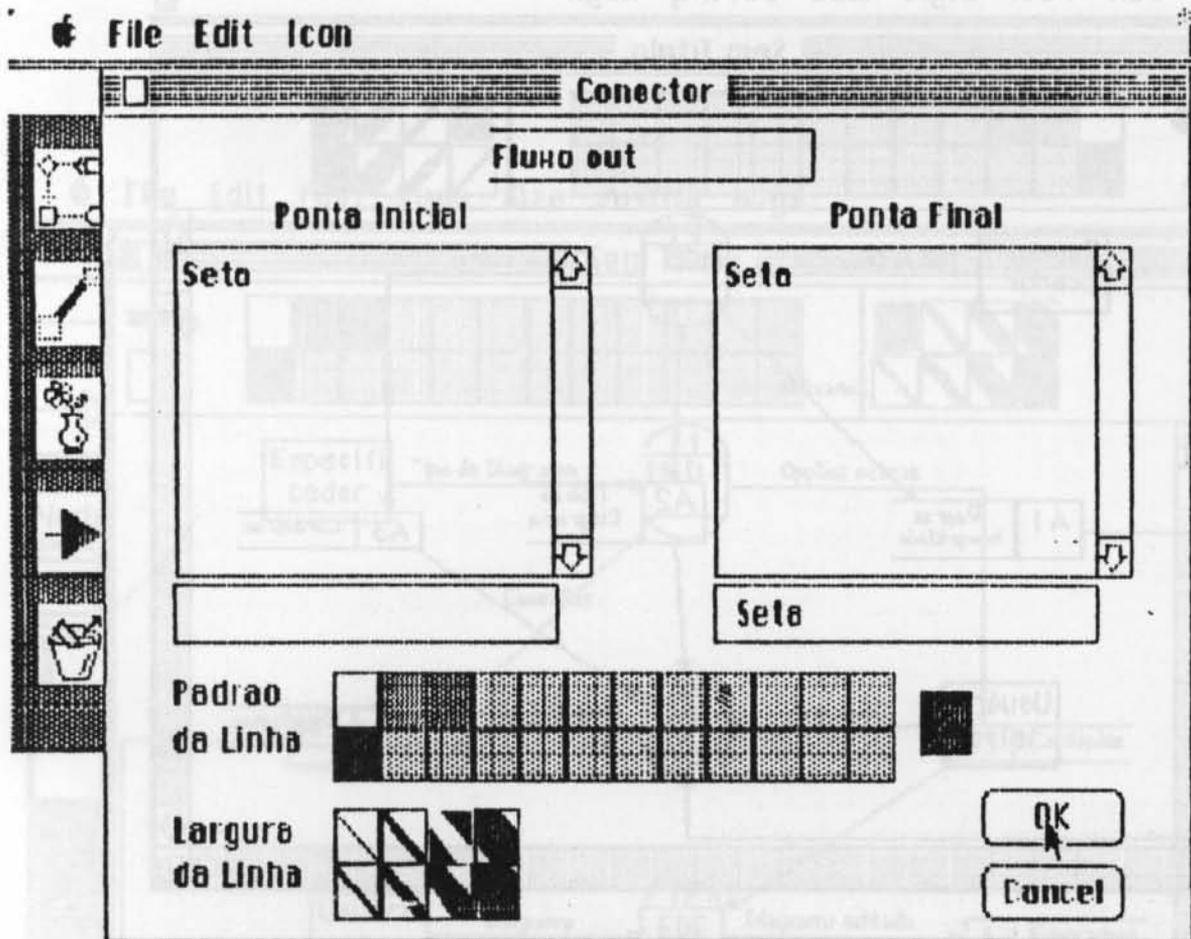


Figura 4 - Edição do arco "Fluxo Out" no MED. A figura mostra o MED no estado de edição de arcos. O arco "fluxo out" está sendo construído com uma ponta denominada "Seta", com o padrão de preenchimento preto.

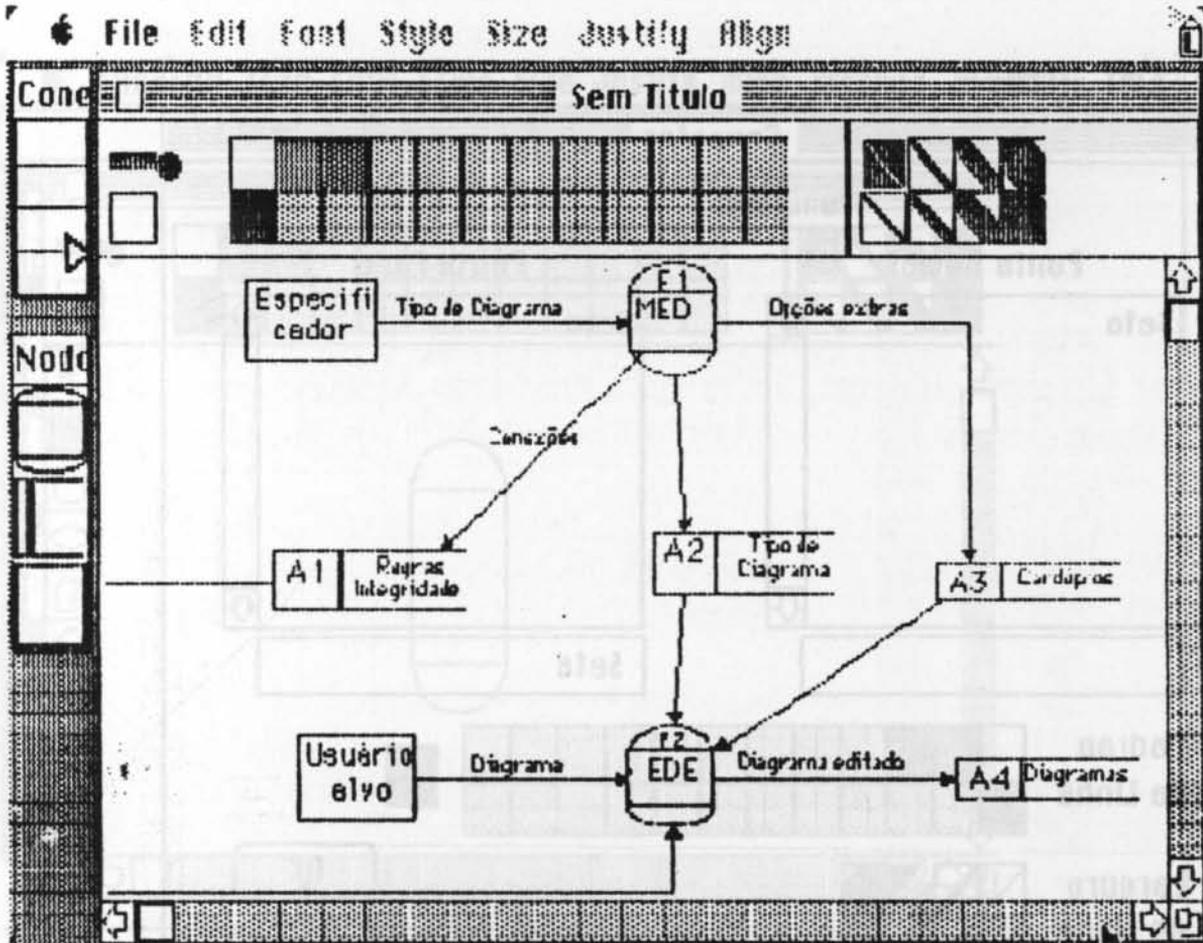


Figura 5 - Edição de um fluxo de dados no editor diagramático específico (EDE). A figura mostra três janelas: a "sem título" onde são editados os diagramas, a "conexões" onde são mostrados os tipos de arcos disponíveis no EDE e a "nodo" onde são mostrados os tipos de nodos. Os tipos de nodos e arcos foram construídos pelo MED. A janela "Cone" mostra os tipos de conectores disponíveis para o usuário. A janela "Nodo" mostra os tipos de nodos disponíveis. Como DFD não possui decorações a janela "Decorações" não é mostrada. O usuário pode trocar o padrão de preenchimento dos objetos ou a largura e o padrão da linha. No diagrama são mostrados arcos formados por mais de um segmento. Inicialmente as ligações são realizadas com um segmento, mas o usuário pode editar o arco com quantos segmentos desejar. O usuário "navega" pelo diagrama através das barras de rolagens (*scroll bars*). O diagrama mostra vários textos associados tanto a nodos quanto a arcos, com tamanhos diferentes. O EDE permite que estes textos sejam formatados com estilos, tamanhos, alinhamentos e fontes variados.

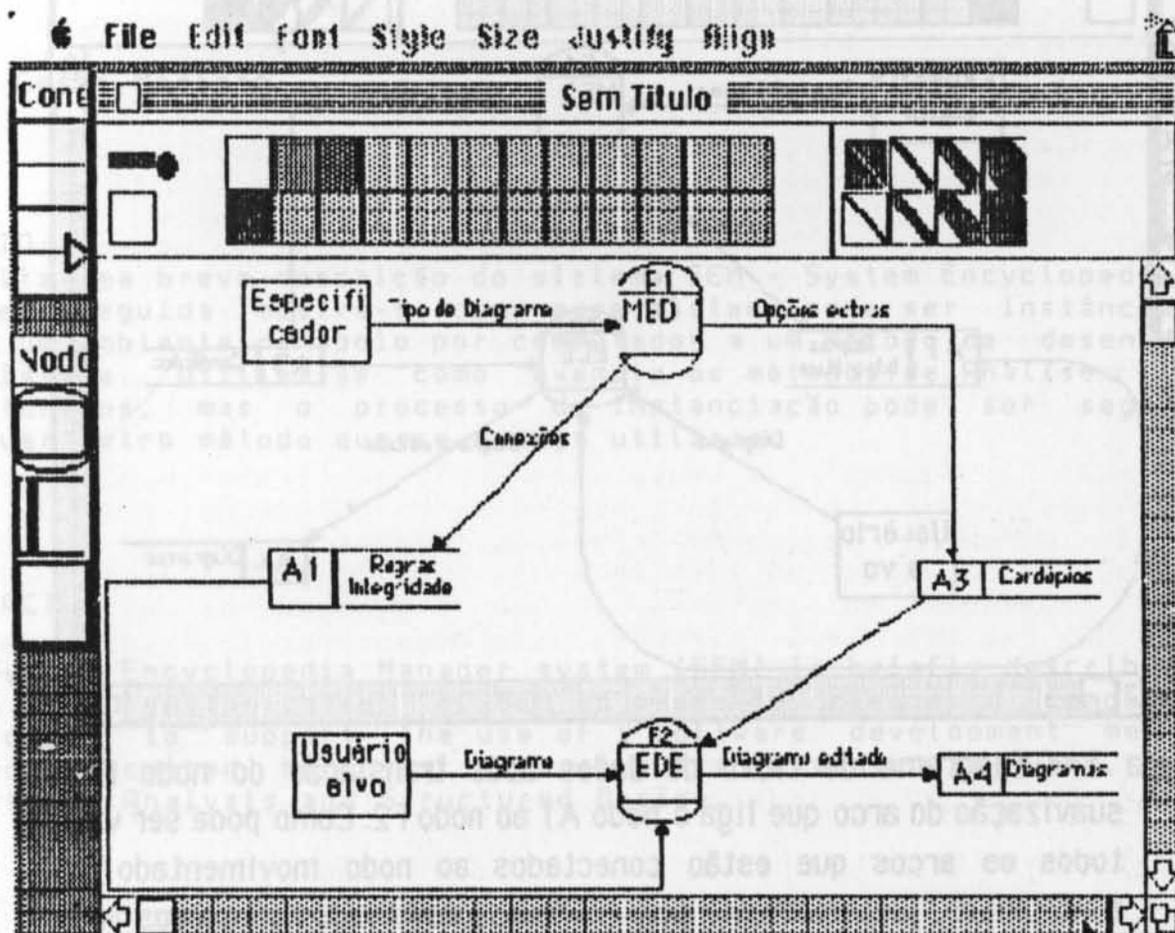


Figura 6 - Diagrama de fluxo de dados após exclusão do nodo A2. Todos os arcos e textos associados ao nodo A2 também são eliminados. O usuário poderá solicitar a exclusão de todos os refinamentos filhos do nodo A2.

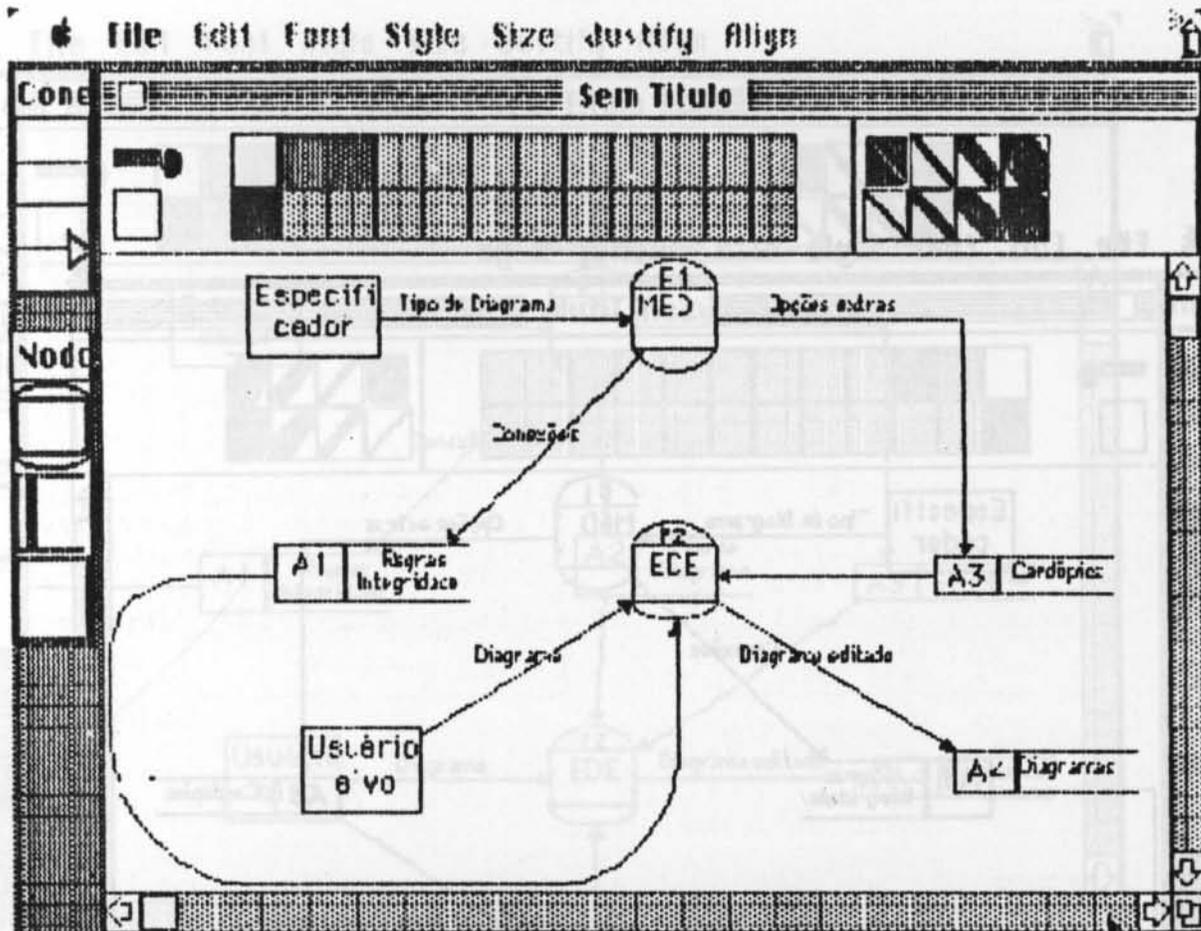


Figura 7 - Diagrama de fluxo de dados após translação do nodo F2 e suavização do arco que liga o nodo A1 ao nodo F2. Como pode ser visto todos os arcos que estão conectados ao nodo movimentado são atualizados, juntamente com os textos associados tanto aos arcos como ao nodo. A suavização do arco foi realizada para exemplo.

Figura 5

Os tipos de nós e arcos locais construídos pelo EDE são mostrados na janela "Tipos de Nós e Arcos". O EDE também suporta nós e arcos de conexão para o diagrama de dados de fluxo de dados. O EDE não possui "Nodo" mostra os tipos de nós disponíveis. Como DFD não possui decorações a janela "Decorações" não é mostrada. O usuário pode trocar o padrão de preenchimento dos objetos ou a largura e o padrão da linha. No diagrama são mostrados arcos formados por mais de um segmento. Inicialmente as ligações são realizadas com um segmento, mas o usuário pode editar o arco com quantos segmentos desejar. O usuário "navega" pelo diagrama através das barras de rolagens (scroll bars). O diagrama mostra vários textos associados tanto a nós quanto a arcos, com tamanhos diferentes. O EDE permite que estes textos sejam formatados com estilos, tamanhos, alinhamentos e fontes variadas.