

UM EDITOR DIAGRAMÁTICO PARA DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO NA INTERNET

Leandro Bento Pompermaier B.Sc.

Roberto Tom Price Eng. M.Sc., D.Phil

Universidade Federal do Rio Grande do Sul – Instituto de Informática
Caixa Postal 15064 – Porto Alegre – RS – Brasil

e-mail: {pomper, tomprice}@inf.ufrgs.br

Resumo

Neste artigo é apresentado EDI, um Editor Diagramático na Internet, que suporta a especificação cooperativo de aplicações. Este editor utiliza tecnologia e funcionalidade dos hiperdocumentos. EDI possui características como: compartilhamento de informações, colaboração por vários autores para a realização de diversas tarefas, vários visões dos dados armazenados, entre outras. A implementação deste editor foi realizada na linguagem de programação Java, e projetada de forma genérica para permitir a criação de editores de diversas notações.

Palavras Chaves: Engenharia de Software, Ambientes de Desenvolvimento de Software, Ferramentas CASE, WWW

Abstract

This paper describes EDI, a Diagrammatic Editor on the Internet, that supports cooperative specification of applications. This editor uses hyperdocument technology and functionalities. EDI offers features such as information sharing, collaboration between several authors, multiple views of the data conceptual stored, among others. EDI is implemented using Java language and is designed with the intention of being generic to enable the easy creation of specific editors for several diagrammatic notations.

Keywords: Software Engineering, Software Development Environment, CASE Tools, WWW

1. Introdução

Ambientes de desenvolvimento de *software* (ADS) auxiliam os desenvolvedores de sistemas nas diversas etapas do ciclo de desenvolvimento, pela integração de ferramentas utilizadas e pelo gerenciamento das várias tarefas associadas a este ciclo [DAR87]. Estes ambientes simplificam o processo de desenvolvimento de *software*, fornecendo convenções uniformes para análise, codificação, testes, etc.

O conceito de ADS surgiu nos anos 70, buscando combinar técnicas, métodos e ferramentas com o objetivo de prover um meio através do qual o desenvolvedor pudesse ter o

apoio na construção do *software* [GIM94][PER91]. Esta forma de tratar o processo de desenvolvimento, como se entende hoje, envolve o suporte a atividades individuais e ao trabalho em grupo, ao gerenciamento do projeto, ao aumento da qualidade geral, ao aumento da produtividade, etc.

Com o surgimento de novas tecnologias, como o ambiente *World Wide Web* (WWW), os sistemas de informação deverão satisfazer requisitos mais abrangentes, utilizando arquiteturas de dados mais complexas, envolvendo visualizações por *browser*, servidores de documentos e servidores de bases de dados, sistemas de informações integrados ao *workgroup* das organizações e trabalho em grupo na geração dos documentos. Estes Sistemas de Informação oferecem através destas interfaces documentos que são sucessivamente manipuladas por diferentes usuários e permitem, a estes usuários, diferentes visualizações das informações que os constituem.

O desenvolvimento de Sistemas de Informação na Internet (SII) requer a modelagem de diversos componentes como, por exemplo, a modelagem estrutural, visual e dos caminhos de navegação dos documentos que os constituem, bem como a modelagem das bases de dados e dos aspectos de atualização e acesso à bases de dados a partir dos documentos, que conterá as informações armazenadas, bem como das transações (casos de uso) e detalhamento dos processos.

Para modelar todos estes aspectos são necessárias ferramentas de edição destes modelos, ferramentas de consistência entre os modelos, ferramentas de transformação e dicionários de dados, constituindo, assim, os componentes de análise e projeto de um ambiente de desenvolvimento de *software*.

Considera-se que para desenvolver SII's seja conveniente que o ambiente de desenvolvimento de *software* também esteja baseado na Internet. Um ambiente de desenvolvimento de *software* na Internet é um ambiente de desenvolvimento que para suportar o ciclo de desenvolvimento de *software* emprega mecanismos de cooperação entre os diferentes usuários, podendo, assim, formar *workgroups* para o desenvolvimento de diversas tarefas. O dicionário de dados, como base de dados comum a todos os usuários implementa um dos mecanismos de compartilhamento de informações. Este ambiente pode garantir a coerência e a consistência das informações durante todo o ciclo de desenvolvimento, permitindo, também, visualização destas informações através de diferentes notações diagramáticas, conforme as preferências dos usuários. Ambientes de desenvolvimento de *software* na Internet poderão apresentar as seguintes características: cooperação de autoria, coordenação entre usuários, diversas visualizações e segurança.

Neste trabalho apresenta-se EDI, um editor diagramático para modelagem de sistemas de informação na Internet. Este artigo está organizado da seguinte forma: a seção 2 apresenta idéias para a construção de ADS na Internet. A seção 3 apresenta características do EDI. A seção 4 mostra a funcionalidade do editor. Na seção 5 é apresentado o dicionário de dados do ambiente, enquanto a seção 6 apresenta algumas conclusões.

2. Ambientes de Desenvolvimento na Internet

Diversas metodologias foram propostas para a construção de *software* Orientado a Objetos (OMT [RUM91], Booch [BOO91], Fusion [COL94], OOSE [JAC94]). Estas metodologias fornecem descrições de como o *software* deve ser criado e mantido [ORT95].

Em geral, estas descrições estão baseadas em um conjunto de documentos produzidos durante o ciclo de vida, definindo a estrutura e os relacionamentos desses documentos.

O processo de desenvolvimento de *software* envolve uma contínua produção e transformação de notações, partindo-se da especificação de requisitos até a produção de código executável [PEN93][FIN94]. Estas produções e transformações podem ser suportadas por ferramentas, que constituem os ambientes de desenvolvimento de *software*. Na produção de *software*, várias ferramentas e notações são utilizadas para atender as fases de seu desenvolvimento[BRO92]. O que ocorre com estas ferramentas é o isolamento ou independência uma das outras, dificultando o relacionamento entre os documentos utilizados e/ou gerados durante este processo.

Tendo em vista a popularização da Internet, principalmente com a tecnologia dos hiperdocumentos, e o surgimento de novas linguagens de programação (Java), se faz cada vez mais necessária a incorporação de tecnologias que suportem o desenvolvimento de *software* de forma distribuída na rede, com a possibilidade dos desenvolvedores atuarem nas diversas etapas existentes do ciclo de desenvolvimento do *software* gerando e trocando suas informações (documentos) entre si como se estivessem num mesmo ambiente local.

Um ADS que possui a Internet como seu principal ambiente de interação permite aos seus usuários a independência de plataforma e de localidade, uma vez que as ferramentas do ambiente podem ser acessadas de qualquer nodo. Além disto, um ADS na Internet apresenta características importantes que o distingue de outros ADS. São elas :

- *Cooperação* : Um ADS na Internet utiliza recursos de comunicação entre os usuários para que estes possam trocar mensagens, a fim de executar uma determinada tarefa ou trocar experiências [REI95]. Os principais meios de comunicação são : *eletronic-mail (e-mail)* e *chat*. Com isto podem ser compartilhadas ferramentas, informações, diagramas, etc. Outra forma de cooperação é o compartilhamento de informações através da base de dados ou através de documentos HTML (*HyperText Markup Language*).
- *Coordenação* : A capacidade de gerenciar as interações dos usuários, controlando o fato de diversas pessoas estarem trabalhando nos mesmos objetos ao mesmo tempo, e também de integrar as contribuições individuais de cada participante num único produto.
- *Segurança* : A segurança consiste em garantir que apenas usuários autorizados terão acesso à base de dados comum. Diferentes tipos de usuários possuem diferentes visões dos dados armazenados e diferentes níveis de acesso. Além dos mecanismos de segurança implementados pelos sistemas de gerenciamento de banco de dados, a Internet provê segurança através dos *browsers* e dos mecanismos existentes na linguagem de programação Java.
- *Autoria* : A autoria consiste em fornecer mecanismos que possibilitem o planejamento da atividade do grupo de trabalho e a representação da mesma no ambiente compartilhado. A Internet facilita o trabalho de autoria, principalmente, pela visualização dos documentos através dos *browsers* e pelo fácil acesso as informações.

Gestão do Dicionário de Dados : repositório de todas as informações do projeto, modelagem, implementação, testes, etc. O ADS deve gerenciar o acesso a base de dados (dicionário de dados) garantindo a consistência e segurança dos dados. Com o dicionário de dados disponível na Internet faz com que os usuários não se preocupem com a localização das informações, principalmente se não estiverem em seu ambiente de trabalho.

- *Consistência dos Modelos* : O ADS deve garantir a consistência dos modelos com relação ao dicionário de dados. Um dicionário de dados integra todos os modelos das notações. A Internet, através dos mecanismos de compartilhamento das informações e coordenação das tarefas dos usuários, garante a consistência dos modelos.
- *Adaptação ao Usuário* : Diversas visões de um mesmo conjunto de dados, possibilitando a adaptação do ambiente às características do usuário. A Internet auxilia na adaptação ao usuário através da homogeneidade de interfaces, proporcionadas pelos *browsers*.
- *Interação Global* : A Internet é, sem dúvida, o maior sistema aberto existente. Com ele diversos usuários podem "navegar" em busca de informações disponibilizadas através das páginas Web. Esta interação global permite um melhor compartilhamento de bibliotecas de componentes, ampla exposição dos produtos gerados nas fases de desenvolvimento, o acesso a *frameworks*, compartilhamento de resultados e experiências.

Adicionalmente um ADS na Internet deve possuir características comuns a outros ADS como, por exemplo, geração de código e engenharia reversa.

2.1 Uma Proposta de ADS na Internet

Um ADS na Internet deve possuir as características citadas anteriormente. Deverá possuir, para tal, ferramentas (editores) capazes de auxiliar o desenvolvimento cooperativo de *software*. Um protótipo de editor cooperativo, desenvolvido em Java é apresentado nas seções subsequentes.

O ADS na Internet oferece homogeneidade de interfaces, conseguida através das visualizações nos navegadores. Este ambiente possui um dicionário de dados armazenado em uma base de dados no servidor. Esta base de dados implementa um modelo conceitual.

Os editores deste ambiente interagem com os usuários através de visões específicas da base de dados. Por exemplo, editor de cenários visualizam objetos e troca de mensagens entre eles. Editor de objetos visualiza a estrutura de herança e associação entre classes. Editor de estados visualiza as transformações de estado dos objetos de uma classe.

Uma mesma visão pode ter diferentes apresentações, conforme o desejo do usuário. A visão do modelo de objetos pode ser vista com letras em negrito e fonte *Arial* por um usuário e itálico e *Times New Roman* por outro.

Pode-se especificar que usos de documentos podem requerer visões específicas. Algumas visões são sobre sub-conceitos específicos (por exemplo a modelagem dinâmica de objetos), outras visões referem-se a forma de visualização de conceitos (por exemplo alguns

usuários preferem ver diagramas de objetos com a notação de Rumbaugh [RUM91], outros com a notação do Booch [BOO91]). Detalhes das visões que são apenas referências de apresentação dos usuários (por exemplo tipo da fonte, tamanho da fonte e cor) são especificadas nas anotações.

O ADS suporta trabalho individual ou cooperativo. A cooperação pode ser de várias formas, como troca de mensagens através de e-mail e chat, compartilhamento de informações através dos modelos visualizados pelos usuários ou na forma de anotações sobre modelos que estão sendo criados. Por exemplo, um usuário cria um modelo na sua visão e o envia a outros usuários, que o podem criticar e sugerir modificações, utilizando anotações sobre o modelo proposto. Estas anotações podem ser na forma de inclusão, remoção, substituição ou comentário

Projetos possuem diversos desenvolvedores que podem estar distribuídos em diversas localidades e que podem realizar tarefas distintas sobre os documentos, ou mesmo cooperar na realização de uma mesma tarefa. O ambiente distribuído suporta este tipo de trabalho, controlando o acesso (prioridades de atualizações, autorizações, gestão de visões e outras) e gerenciando os documentos.

3. Funcionalidades e Características do EDI

O editor diagramático para a Internet (EDI) é um editor de diagramas, com características próprias, tais como: vários usuários podem realizar diversas tarefas, trocando informações e compartilhando uma única base de dados; cada usuário pode visualizar de forma específica um mesmo conjunto de dados; independência de plataforma, uma vez que a ferramenta é executada sobre navegadores Web (Netscape, Internet Explorer, Mosaic, etc.)

Vários usuários acessando a ferramenta ao mesmo tempo, e requerendo visualizações diversas sobre a mesma base de dados, requer do sistema capacidade de gerenciar estas visões e a atualização da base de dados.

No protótipo o modelo é representado pela base conceitual. Esta base é gerenciada por um sistema de banco de dados e representa os conceitos utilizados pelas representações das notações. Na seção 5 é apresentado detalhes do modelo conceitual (dicionário de dados) As visões são dependentes de como cada usuário visualiza as informações da base de dados. Cada visão é apresentada pelo editor específico da visão. A atualização da base de dados e das diversas visões se dá de forma consistente, pois, quando o modelo é alterado (base de dados), automaticamente as visões agregadas são notificadas da alteração.

Quando uma determinada tarefa é inicializada, somente um usuário receberá a especificação (usuário proprietário). Analisando a especificação da tarefa, este usuário deverá requisitar o bloqueio dos conceitos que, possivelmente poderão sofrer alterações. Estes conceitos variam de acordo com o editor utilizado para o desenvolvimento da tarefa como, por exemplo, se o usuário utilizar o editor de objetos, as classes, atributos, métodos, associações, agregações e heranças utilizadas na modelagem da tarefa deverão ser bloqueados para que as alterações realizadas sobre estes conceitos não tornem os modelos extraídos a partir destes conceitos inconsistentes.

Cada usuário proprietário poderá criar um grupo de trabalho ou grupo de discussão para que haja troca de experiências entre vários usuários. Este usuário (usuário proprietário)

enviará parte ou todo modelo criado para um ou vários *usuários colaboradores*. Estes usuários receberão, além da visão criada pelo usuário proprietário um submodelo conceitual, que é cópia do dicionário de dados contendo os conceitos manipulados pelo modelo em questão. Cada usuário, tanto proprietário como colaborador, possuirá um *profile* contendo características específicas de apresentação dos modelos como, por exemplo, tipo e cor de letra.

Os usuários, tanto proprietário como colaborador, possuem um conjunto de funções que foram previamente estabelecidas. A tabela 1 apresenta estas funções.

Tipo de Usuário	Funções
Proprietário	<ul style="list-style-type: none"> - Criar visão - Bloquear modelo - Editar visão - Incluir visão no modelo - Desbloquear modelo - Enviar visão para análise dos colaboradores - Agregar anotações ao modelo
Colaborador	<ul style="list-style-type: none"> - Receber visões - Anotar visões - Devolver visões - Enviar visões para análise de outros colaboradores - Agregar anotações ao modelo

Tabela 1 – Funções dos Usuários

Todas as alterações realizadas pelos usuários (proprietário e desenvolvedor) serão armazenadas em um sistema de *log*. Este sistema será disponibilizado, como um todo, somente para o desenvolvedor proprietário. Assim este poderá escolher e/ou visualizar as alterações que melhor representam o desenvolvimento da tarefa. Ou seja, a forma de manter controle de versões das visões é implementada pelo *log*.

Entre as funcionalidades do editor pode-se salientar o mecanismo de colaboração no desenvolvimento das tarefas. A colaboração pode ser indireta, através do compartilhamento de documentos, isto é, os diversos usuários do editor poderão, através de um sistema de arquivos, compartilhar documentos (diagramas, textos e imagens).

Outra forma é a colaboração direta, através de mecanismos de troca de mensagens e anotações. Os usuários (desenvolvedores) podem trocar mensagens (informações e experiências) através de protocolos de comunicação (correio eletrônico (e-mail) e *chat*) implementados em Java. Estes programas agregados ao ambiente permite a formação de *workgroup* entre usuários ou envolvidos na mesma tarefa ou especialistas em uma determinada área. O mecanismo de *chat* permite uma interação entre todos os usuários do editor ou somente entre os usuários cadastrados no mesmo grupo de trabalho.

A colaboração através de anotações é uma forma de comunicar idéias ou opiniões sobre o conteúdo de um diagrama criado. Um usuário colaborador analisa o modelo recebido, insere suas anotações junto com o diagrama e envia ao usuário proprietário do diagrama. Estas anotações são inseridas com o intuito de aperfeiçoar o documento, e também podem ser

utilizadas como registro das argumentações ocorridas durante a criação do diagrama. Além disso ela serve como um mecanismo de comunicação entre o usuário proprietário e os usuários colaboradores. Esta colaboração pode ser (figura 1): (a) apresentar idéias ou opiniões; (b) propor alterações; e (c) levantar questões. Toda anotação deve ter o identificação do comentador.

Funções de edição semânticas dos modelos permitem ao usuário alterar semanticamente o modelo, realizando troca de domínio de modelagem. Por exemplo, o usuário poderá transformar um atributo em classe e vice-versa, utilizando estas funções e mantendo a consistência com relação ao dicionário de dados.

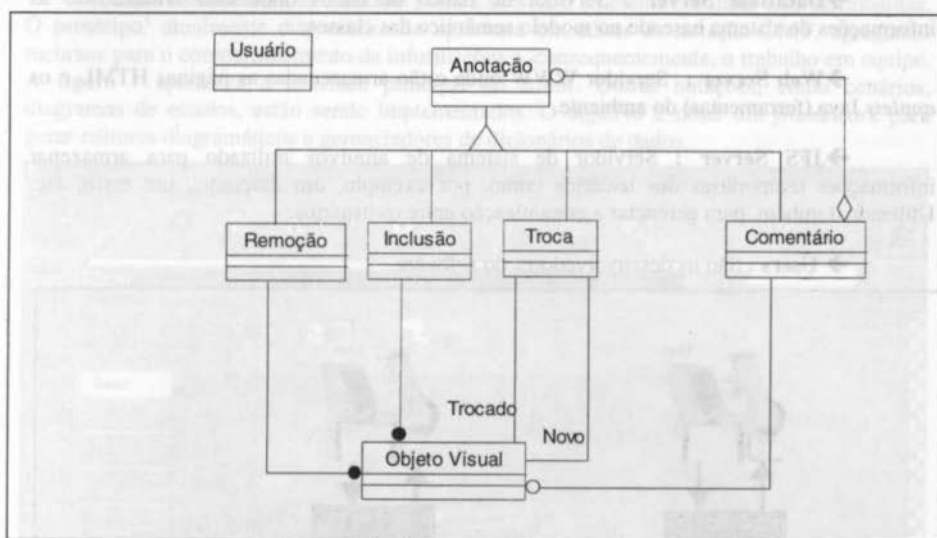


Figura 1 – Modelo de Objetos para Anotações baseado em [SOU97]

4. Aspectos de Implementação do EDI

O editor foi programado como uma *applet* (que executa no cliente) e uma aplicação (que executa no servidor) que gerencia os usuários. Para a implementação do EDI, sendo este uma ferramenta gráfica, foi utilizado o *framework Hotdraw* [JOH92] e para a solução de problemas relativos a implementação gráfica de alguns componentes utilizados pelas notações que o EDI suporta foram utilizados os padrões de projeto Observer e Composite [GAM95]. Foi utilizado, também, o *framework Java File System (JFS)* para a construção e gerenciamento do sistema de arquivos das visões manipuladas pelos clientes

Cada notação tem um EDI específico. Está sendo construído EDI específicos, utilizando as classes definidas em EDI anteriores para chegar a definição de um *framework* para a criação de editores diagramáticos.

A figura 1 fornece uma visão geral da arquitetura do EDI. O ambiente proposto possui três camadas : *Database Server*, *Web Server* e *JFS Server* (figura 2). Estas três camadas podem estar presentes em uma única máquina, ou distribuídas em várias máquinas.

O usuário acessa o servidor Web (*Web Server*) na *Universal Resource Locator* (URL) do ambiente. Ao carregar a página HTML irá também carregar o *applet Java*, que consiste no EDI, podendo criar, alterar ou anotar modelos diagramáticos.

O EDI possui a seguinte arquitetura :

→ **DataBase Server** : Servidor de banco de dados onde está armazenado as informações do sistema baseado no modelo semântico das classes;

→ **Web Server** : Servidor WWW onde estão armazenadas as páginas HTML e os *applets Java* (ferramentas) do ambiente.

→ **JFS Server** : Servidor de sistema de arquivos utilizado para armazenar informações temporárias dos usuários como, por exemplo, um diagrama, um texto, etc. Utilizado, também, para gerenciar a comunicação entre os usuários.

→ **Users** : São os desenvolvedores do *software*.

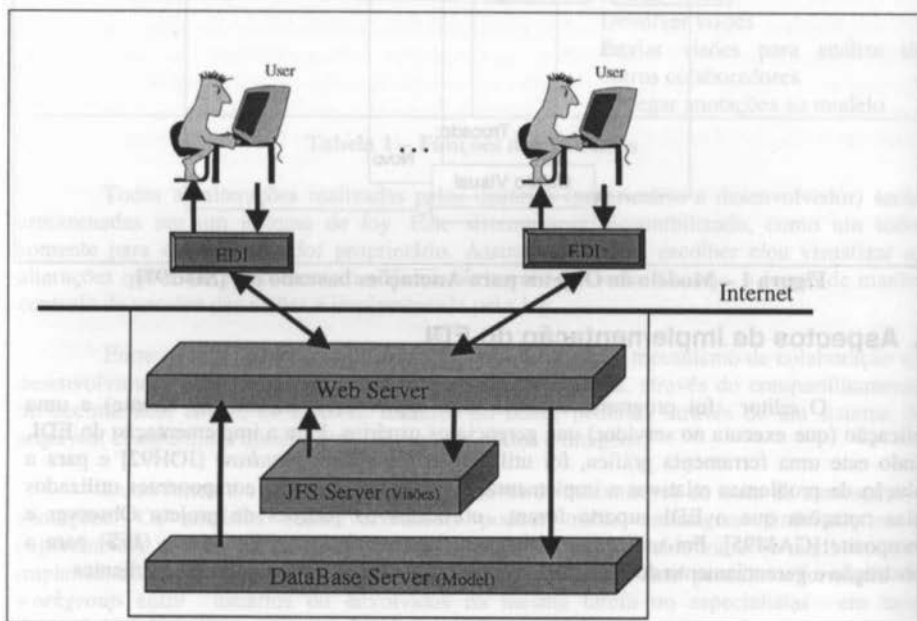


Figura 2 - Arquitetura do Editor Diagramático para Internet

5. Exemplo de Utilização do EDI

O protótipo foi implementado na linguagem Java utilizando a arquitetura MVC e sistema de arquivos JFS (*Java File System*). Para executar a ferramenta é necessário o acesso

a URL do ambiente e acessar suas respectivas páginas HTML, utilizando um navegador que suporte a linguagem Java, isto é, a execução dos *applets*.

Antes do usuário acessar as ferramentas do ambiente, este deve ser cadastrado no sistema pelo usuário *root*. Este cadastramento define o grupo de trabalho do usuário, sua área de armazenamento local, *username* entre outras características. Além do cadastro dos usuários do ambiente, é possível o cadastro de impressoras e o monitoramento das conexões dos usuários. Cada usuário cadastrado recebe uma senha que lhe permitirá o acesso nas ferramentas do ambiente.

Uma vez cadastrado, o usuário poderá acessar o ambiente e utilizar suas ferramentas. O protótipo¹ atualmente disponibilizado apresenta um editor de notação OMT, agregando recursos para o compartilhamento de informações e, conseqüentemente, o trabalho em equipe. A figura 1 apresenta a interface principal do editor. Outras notações, como cenários, diagramas de estados, estão sendo implementados. O objetivo é obter um *framework* para gerar editores diagramáticos e gerenciadores de dicionários de dados.

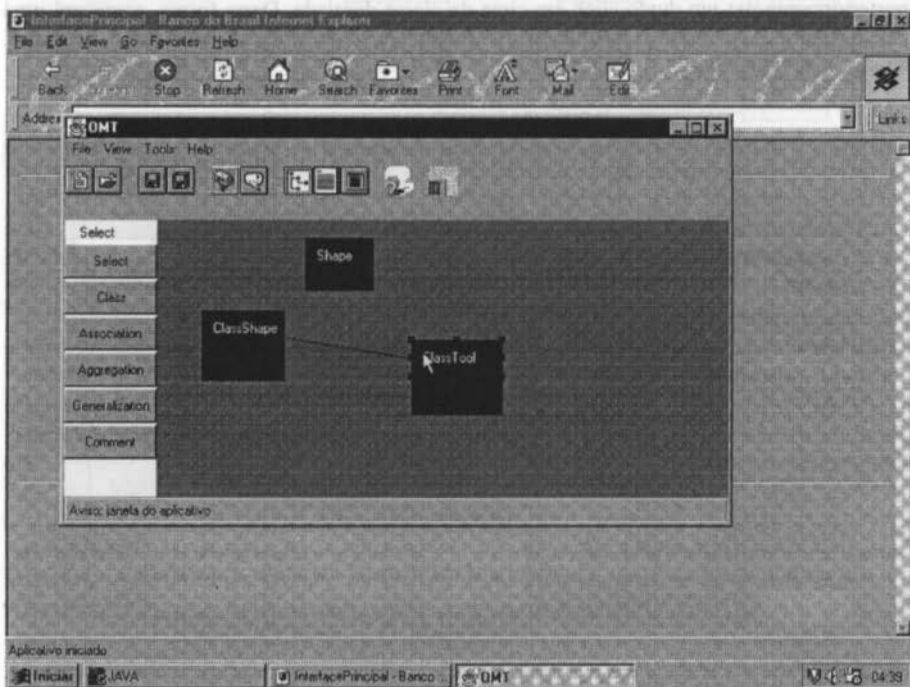


Figura 3 - Visão geral da Ferramenta de Modelagem

O editor é dividido em :

→ menu principal : com as opções *File* (manipulação de arquivos), *View* (visualização do modelo, quanto a notação, e visualização das tarefas em andamento e por

¹ Atualmente o protótipo do EDI está disponível na seguinte URL : <http://www.inf.ufrgs.br/~pomper/EDI/>

realizar), *Tools* (gerador de linguagem Java), *Help* (ajuda ao usuário com relação aos recursos técnicos do editor);

→ barra de ferramentas horizontal : apresenta ferramentas para o armazenamento e compartilhamento das informações individuais e ferramentas para acesso aos recursos de troca de informações (*mail e chat*);

→ barra de ferramentas vertical : ferramentas para edição da notação. No protótipo é disponibilizado ao usuário ferramentas para acrescentar classes, relacionamentos entre as classes (herança, agregação e associação) e, também, acrescentar comentários e anotações ao modelo, tanto genéricos como associados a algum componente gráfico;

→ área de desenho : onde será projetado o sistema de informação.

Para alterar características de classes como, por exemplo, nome, tipo (abstrata, final, publica ou privada), incluir atributos e métodos deve-se selecionar a ferramenta *Select* e posteriormente dar um duplo click em cima da classe desejada. Desta forma, aparecerá uma janela (figura 2) contendo as informações da respectiva classe, podendo estas serem incluídas, alteradas ou removidas.

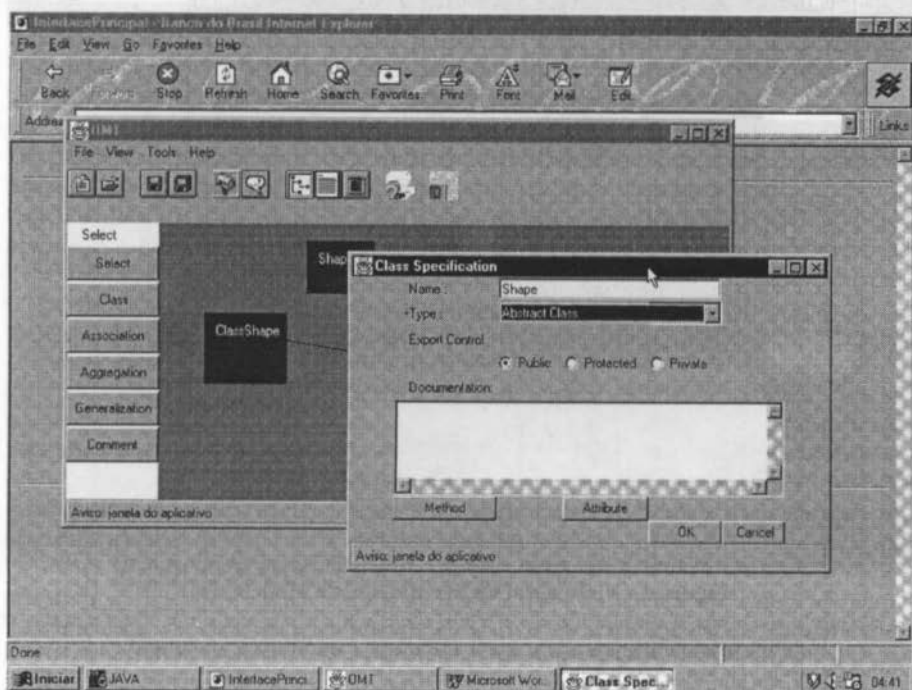


Figura 4 - Janela de Especificação da Classe

6. Modelo Semântico do Dicionário de Dados

Para o modelo de dados, utilizado para descrever o modelo conceitual do domínio da aplicação, foi utilizada a notação OMT (*Object Modeling Technique*) [RUM91]. Este modelo fornece uma estrutura semântica que facilita a descrição do domínio e a representação dos conceitos utilizados pelas notações utilizadas nas ferramentas implementadas.

O editor possui uma base de dados comum (dicionário de dados do ADS), onde estão armazenadas as informações dos sistemas. Esta base de dados deverá representar todos os conceitos manipulados pelas ferramentas, isto é, apresentar em modelo conceitual (figura 3) todos os conceitos presentes nas diversas visões que as ferramentas específicas manipulam.

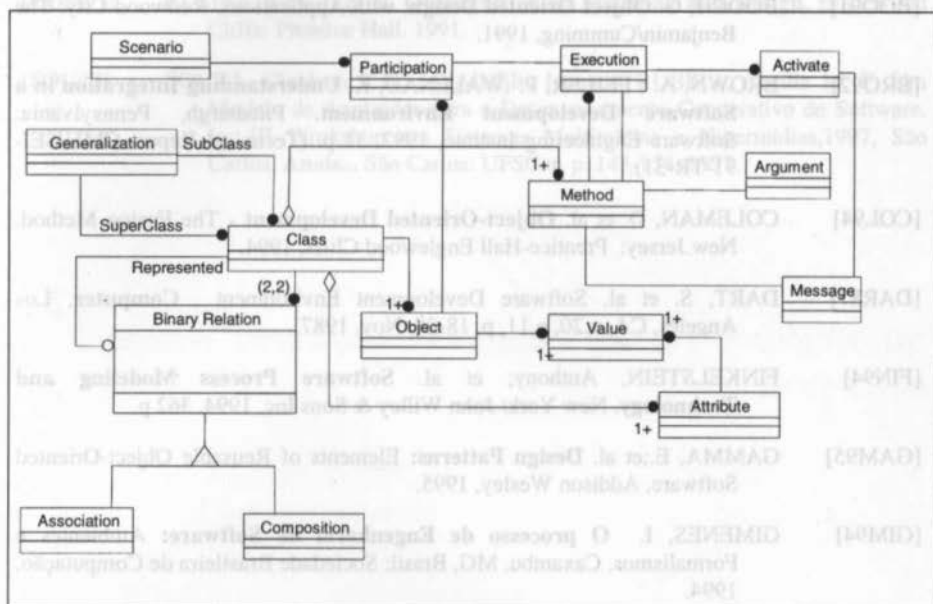


Figura 5 - Modelo Parcial de Objetos Semânticos do Ambiente

A partir deste modelo conceitual cada editor representará parte destes conceitos de forma gráfica, de acordo com a notação utilizada. Cada notação deverá ter um editor próprio, especializado do *framework* de edição genérico. Cada editor manipula um submodelo conceitual.

7. Conclusões

Os resultados obtidos com esta pesquisa, demonstram a viabilidade de se combinar um ambiente aberto (Internet) com o desenvolvimento cooperativo de *software*. A cooperação na especificação de sistemas de informação é obtida através do compartilhamento das visões e das anotações adicionadas às visões pelos colaboradores.

A abordagem de diferentes visões de um mesmo modelo possibilita o uso integrado de diferentes notações para a especificação de sistemas de informação por diferentes usuários. Esta abordagem de múltiplas visões é conseguida através da implementação da arquitetura MVC, permitindo o EDI a trabalhar com um mesmo modelo semântico.

O trabalho descrito faz parte de um projeto que envolve a especificação de um *framework* para a construção de editores diagramáticos. Esta especificação deverá ser alcançada através do processo evolutivo que está sendo empregado na construção de EDI's, analisando cada risco envolvendo as características de editores diagramáticos colaborativos para Internet.

Referências Bibliográficas

- [BOO91] BOOCH, G. **Object Oriented Design: with Applications**. Redwood City: The Benjamin/Cumming, 1991.
- [BRO92] BROWN, A. ; FEILER, P. ; WALLNAU, K. **Understanding Integration in a Software Development Environment**. Pittsburgh, Pennsylvania: Software Engineering Institute, 1992. 31 p. (Technical Report CMU/SEI-91-TR-31).
- [COL94] COLEMAN, D. et al. **Object-Oriented Development - The Fusion Method**. New Jersey: Prentice-Hall Englewood Cliffs, 1994.
- [DAR87] DART, S. et al. Software Development Environment . **Computer**, Los Angeles, CA, v.20, n.11, p. 18-28, Nov. 1987.
- [FIN94] FINKELSTEIN, Anthony; et al. **Software Process Modeling and Technology**. New York: John Willey & Sons Inc. 1994. 362 p.
- [GAM95] GAMMA, E.; et al. **Design Patterns: Elements of Reusable Object-Oriented Software**. Addison Wesley, 1995.
- [GIM94] GIMENES, I. **O processo de Engenharia de Software: Ambientes e Formalismos**. Caxambu, MG, Brasil: Sociedade Brasileira de Computação, 1994.
- [JAC94] JACOBSON, I. et al. **Object-Oriented Software Engineering - A Use Case Driven Approach**. Addison-Wesley Publishing Company, 1994.
- [JOH92] JOHNSON, R. E. **Documenting Frameworks using Patterns**. In: Object-Oriented Programming Systems, Languages and Applications Conferende - OOPSLA, 1992, Vancouver. **Proceedings...Vancouver**.
- [ORT95] ORTIGOSA, Alvaro Manuel. **Proposta de um Ambiente Adaptável de Apoio ao Processo de Desenvolvimento de Software**. Porto Alegre: CPGCC da UFRGS, 1995. (Dissertação de Mestrado).
- [PAR94] PARCPLACE System Inc. **Visual Works Reference Manual**. CA: ParcPlace System Inc. , 1994.

- [PEN93] PENEDO, M. Process-based Software Engineering Environments (PSEE). In: BRAZILIAN SYMPOSIUM OF SOFTWARE ENGINEERING, 7., 1993, Rio de Janeiro. **Anais...** Rio de Janeiro: PUC do Rio de Janeiro, 1993. (palestra convidada).
- [PER91] PERRY, D.; KAISER, G. Models of Software Development Environments. **Transactions on Software Engineering**. New York, v.17, n.3, p.283-295, Mar 1991.
- [REI95] REIS, Rodrigo Quites. **Desenvolvimento Cooperativo de Software: Ferramentas e Metodologias**. Porto Alegre : CPGCC-UFRGS. 1995. (Trabalho Individual)
- [RUM91] RUMBAUGH, J. et al. **Object-Oriented Modeling and Design**. Englewood Cliffs: Prentice Hall. 1991.
- [SOU97] SOUZA, Cleidson R.B. ; WAINER, Jacques; RUBIRA, Cecília M. F. Um Modelo de Anotações para o Desenvolvimento Cooperativo de Software. In: III Workshop em Sistemas Multimídias e Hipermídias, 1997, São Carlos. **Anais...** São Carlos: UFSCar, p. 143-154, 1997.

Sessão Técnica: Técnicas para Produção de Software