

Ferramentas de apoio à UML: um modelo para avaliação baseado em requisitos funcionais e não-funcionais

JULIANE FORESTI¹, LIS ÂNGELA DE BORTOLI¹

¹Instituto de Ciências Exatas e Geociências – Universidade de Passo Fundo (UPF)
forestirs@yahoo.com.br, lis@upf.br

Resumo. Este artigo apresenta um modelo para avaliação de ferramentas que oferecem suporte à UML, baseado em requisitos funcionais, não-funcionais, bem como em normas de qualidade.

Palavras-chave: UML, ferramentas de apoio, qualidade, requisitos, CASE

UML support tools: an evaluation model based on functional and non-functional requirements

Abstract. This article presents a model to evaluate UML support CASE tools based on functional requirements, non-functional requirements and quality norms.

Key-words: UML, support tools, quality, requirements, CASE

(Recebido para publicação em 25 de outubro de 2004 e aprovado em 6 de dezembro de 2004)

1. Introdução

O uso de software de apoio à modelagem é muito importante por dois motivos: primeiro porque os modelos começarão a ficar tão longos que a folha de papel ficará pequena, segundo porque é uma ótima maneira de checar as associações entre os modelos. [5]. O principal objetivo da tecnologia CASE é separar o projeto da implementação do código.

Existem várias ferramentas de modelagem, algumas suportando a UML, porém muitas apóiam um método particular que em si atende a um tipo de projeto. É essencial avaliar os benefícios das ferramentas e suas limitações, evitando problemas posteriores no processo de desenvolvimento. Com um grande número existente de ferramentas CASE, com vários recursos característicos que proporcionam, é necessário conhecer quais delas apóiam melhor a UML. Entretanto para que isto seja possível é preciso fazer uma verificação das ferramentas disponíveis e avaliar suas principais características.

Atualmente existem várias ferramentas de apoio à UML. O *site Objects by Design* [7] apresenta, aproximadamente, 115 ferramentas com informações sobre seus fabricantes, versões, datas, plataformas e preços (nem sempre disponível).

O presente artigo apresenta um modelo para avaliação de ferramentas de suporte à UML, baseado em requisitos funcionais, não-funcionais, bem como em normas de qualidade. Inicialmente apresenta-se as categorias definidas no modelo de avaliação, bem como os critérios que formam cada categoria. Apresenta-se a seguir o estudo de caso que foi definido e realizado a fim de validar o modelo proposto. Posteriormente, são apresentados os principais resultados obtidos com o estudo de caso, algumas observações complementares constatadas na avaliação e finalmente a conclusão.

2. O Modelo Proposto

O site anteriormente mencionado [7] apresenta, também, uma lista comentada de itens que devem ser considerados nas ferramentas. Porém, a lista não aponta quais tipos de respostas devem ser buscados e

nem uma organização precisa dos itens (classificação), o que dificulta, em parte, sua aplicação. Além disso, os itens não são relacionados com normas de qualidade. Sendo assim, a finalidade do modelo proposto é estabelecer um conjunto de critérios, baseados em normas de qualidade e classificados em requisitos funcionais e não-funcionais, que permitam a avaliação de ferramentas de apoio à UML. Após a avaliação, segundo os critérios propostos, as informações obtidas poderão auxiliar o futuro usuário na escolha da ferramenta de apoio à UML mais adequada as suas necessidades.

Os critérios definidos no modelo permitem especificar como as ferramentas trabalham, se realmente executam o que seu fabricante anuncia e se atendem aos requisitos dos usuários com qualidade. Porém, não é objetivo do modelo indicar qual ferramenta é a melhor. A opção será feita pelo usuário a partir dos resultados obtidos.

Para uma melhor definição dos critérios, investigou-se algumas normas de qualidade existentes. As normas de qualidade investigadas foram: ISO/IEC 9126 - Engenharia de Software - Qualidade de produto de software, ISO/IEC 14598 - Tecnologia de informação – Guias para avaliação de produto de software, NBR ISO/IEC 12119 - Tecnologia de informação - Pacotes de software - Testes e requisitos de qualidade [4] e ISO/IEC 14102 - Tecnologia de informação – Orientação para avaliação e seleção de ferramentas CASE.

A lista de critérios para avaliação das ferramentas foi criada para direcionar a avaliação. Os 57 critérios foram divididos em funcionais e não funcionais e apresentam duas formas de resposta como mostra a Tabela 1. Para uma melhor organização dos critérios, os mesmos foram classificados em categorias que servem para agrupar os critérios que possuem características semelhantes. Apresenta-se, a seguir, as oito categorias de critérios que foram criadas, bem como os objetivos principais de cada uma delas:

- *Baseados na norma ISO/IEC 9126:* verificar nas ferramentas a existência de características relacionadas a esta norma;
- *Baseados em características de hardware e software:* verificar em quais plataformas de hardware e software as ferramentas podem ser utilizadas;
- *Baseados no repositório de dados:* verificar quais são as funções do repositório nas ferramentas;
- *Baseados na documentação:* verificar que tipos de documentação acompanham as ferramentas

e se as mesmas auxiliam os desenvolvedores no uso das ferramentas;

- *Baseados no uso da ferramenta com relação ao computador utilizado:* verificar qual o desempenho das ferramentas com relação ao computador utilizado para avaliação;
- *Baseados na UML:* verificar quais funções executadas pelas ferramentas correspondem com as características da UML;
- *Baseados no fornecedor:* possibilitar informações detalhadas sobre o fornecedor, produtos e suporte aos mesmos;
- *Baseados nas necessidades identificadas com a utilização das ferramentas:* verificar se as ferramentas atendem as necessidades dos usuários.

Tabela 1 – Tipos de Resposta

Critério	Tipo de resposta	Código
É possível gerar código fonte a partir da modelagem ?	Sim/Não/Em parte, justificando se necessário.	1
Quais os bancos de dados apoiados pela ferramenta?	Resposta descritiva	2

Nas próximas seções serão apresentados os critérios de cada categoria, bem como a motivação da criação de cada uma das categorias. Nas tabelas, a coluna “Resp” refere-se ao tipo de resposta do critério, conforme Tabela 1 e a coluna “Req” (Requisito) corresponde a F (Funcional) ou NF (Não-Funcional).

2.1 Critérios baseados na norma ISO/IEC 9126

Os critérios desta categoria foram criados com base na norma ISO/IEC 9126. O objetivo é descobrir como as ferramentas atendem algumas características desta norma. Foram consideradas as características manutenibilidade (sub-características modificabilidade e analisabilidade), confiabilidade (sub-característica maturidade), funcionalidade (sub-característica interoperabilidade) e apreensibilidade. Nesta categoria foram criados vinte e dois critérios sendo que a Tabela 2 apresenta alguns exemplos.

Tabela 2– Critérios baseados na norma ISO/IEC 9126

Critério	Objetivo	Resp	Req
É possível incluir, excluir, mover, agrupar, desagrupar e redimensionar objetos?	Avalia a possibilidade de manipulação dos objetos na ferramenta depois de criados. O objetivo é verificar se após os elementos serem criados, os mesmos podem ser modificados.	1	F
Possui opções para análise dos diagramas?	Verificar se a ferramenta auxilia o desenvolvedor na depuração da modelagem, analisando todos os elementos criados.	1	F
É possível gerar código fonte a partir da modelagem criada?	Verificar se o usuário poderá gerar código para a aplicação que está sendo desenvolvida.	1	F
É possível fazer engenharia reversa de dados?	Verificar se a ferramenta possui este recurso.	1	F
É possível importar e exportar dados?	Verificar quais ferramentas importam/exportam arquivos do projeto que está sendo criado.	1	NF
Na ferramenta é possível valer-se de informações de outras ferramentas?	Avaliar se é possível utilizar subsídios de outras ferramentas.	1	NF
Os arquivos criados na ferramenta são portáteis para a mesma ferramenta em outra plataforma?	Avaliar se os arquivos criados na ferramenta podem ser utilizados nas versões da ferramenta criadas para outras plataformas.	1	NF
A ferramenta possui restrição quanto ao número de objetos criados pela mesma na modelagem?	Avaliar se a ferramenta possui alguma restrição quanto a criação de elementos na ferramenta.	1	NF
Existe maneira de prevenir falhas originadas por <i>hardware</i> ou <i>software</i> ?	Verificar se a ferramenta possui recuperação de dados por meio de defeitos de hardware e/ou software.	1	NF
A organização gráfica dos elementos visuais promove a compreensão do usuário na utilização da ferramenta?	Avaliar se a ferramenta é fácil de ser manipulada, verificando se o usuário terá facilidade para utilização da ferramenta a partir da visualização dos elementos.	1	NF

2.2 Critérios baseados na UML

Esta categoria é composta de quatro critérios que permitem avaliar a relação da ferramenta com características especiais da UML, como possibilidade de definição de herança múltipla, possibilidade de criação de todos os diagramas propostos pela linguagem e possibilidade de armazenar tarefas intermediárias dos casos de uso. A Tabela 3 apresenta os critérios definidos para esta categoria.

Tabela 3 – Critérios baseados na UML

Critério	Objetivo	Resp	Req
Permite definir tarefas intermediárias em casos de uso?	Verificar se é possível documentar tarefas intermediárias dos casos de uso.	1	F
Permite definição de herança múltipla?	Verificar se a ferramenta possui esta opção no diagrama de classes	1	F
Permite a criação de todos os diagramas propostos pela UML?	Verificar se é possível criar todos os diagramas da UML.	1	F
Qual é a versão da UML suportada pela ferramenta?	Informar ao usuário qual é a versão da UML suportada pela ferramenta.	2	NF

2.3 Critérios baseados no fornecedor

Com os critérios desta categoria é possível obter-se informações mais detalhadas sobre os fornecedores das ferramentas, como tempo de existência, forma de comercialização dos produtos, suporte oferecido, etc. Foram definidos onze critérios e todos correspondem a requisitos não-funcionais. A Tabela 4 apresenta exemplos de critérios definidos para esta categoria.

Tabela 4 – Critérios baseados no fornecedor

Critério	Objetivo	Resp
Há quanto tempo o fornecedor está no mercado?	Informar há quanto tempo o fornecedor está atuando no mercado.	2
O fornecedor comercializa outros produtos?	Verificar se existem outros produtos desenvolvidos pelo fornecedor.	2
Há quanto tempo a ferramenta está disponível?	Verificar há quanto tempo a ferramenta está disponível para utilização.	2
Como é possível adquirir a ferramenta?	Verificar quais são as possibilidades de aquisição da ferramenta.	2
Como é feito o suporte aos clientes?	Verificar como é realizado o atendimento ao usuário.	2
O produto possui alguma certificação de qualidade?	Verificar se a ferramenta possui alguma certificação de qualidade ou se existe alguma preocupação por parte do fabricante quanto a isto.	1

2.4 Critérios baseados nas necessidades identificadas com a utilização da ferramenta

A partir dos seis critérios definidos nesta categoria é possível verificar se as ferramentas atendem as necessidades dos usuários na utilização das mesmas. A Tabela 5 apresenta exemplos de critérios definidos para esta categoria.

Tabela 5 – Critérios baseados nas necessidades identificadas com a utilização da ferramenta

Critério	Objetivo	Resp	Req
É possível gerar histórico das ações executadas?	Avaliar se disponibiliza registros do que é criado.	1	F
É possível abortar/ desfazer ações executadas?	Avaliar se existe controle para determinadas ações, verificando se é possível desfazer ou abortar ações facilitando a criação dos modelos.	1	F
Existe alguma forma de controle e gerenciamento de usuários?	Avaliar se existe controle e gerenciamento de usuários.	1	F
Quais são os conhecimentos mínimos para usar?	Identificar e informar quais devem ser os conhecimentos básico de utilização.	2	NF

2.5 Critérios baseados em características de hardware e software

A partir dos critérios desta categoria é possível verificar as necessidades de hardware e software das ferramentas. Foram definidos cinco critérios nesta categoria e todos correspondem a requisitos não-funcionais. Alguns exemplos são:

- *É possível executar a ferramenta em quais sistemas operacionais?* O objetivo deste critério é identificar para quais sistemas operacionais as ferramentas estão disponíveis. A resposta para este critério é descritiva (tipo 2), ou seja, uma lista de sistemas operacionais.
- *Quais são os requisitos mínimos aconselhados pelo fornecedor para a utilização da*

ferramenta? Busca-se verificar quais os requisitos (memória, espaço em disco, processador, etc) mínimos para a utilização das ferramentas. A resposta informa ao usuário qual deve ser a capacidade mínima do seu computador para que ele possa utilizar a ferramenta satisfatoriamente. A resposta para este critério é descritiva (tipo 2), ou seja, a configuração mínima sugerida pelo fabricante.

- *É preciso uma base de dados específica para a ferramenta?* Com este critério é possível investigar se as ferramentas necessitam de alguma base de dados para a criação dos diagramas. A resposta para este critério é do tipo 1.

2.6 Critérios baseados no repositório de dados

Nesta categoria buscou-se informações que permitissem avaliar as funções dos repositórios de dados das ferramentas. Para atingir este objetivo foram criados três critérios:

- *Existe repositório?* O objetivo deste critério é avaliar de que forma a ferramenta armazena os dados.
- *Através do repositório é possível criar informações (classes, atributos, ligações)?* Com este critério busca-se avaliar se os repositórios das ferramentas permitem a criação dos elementos dos diagramas a partir das informações salvas.
- *É possível recuperar os dados do repositório?* Busca-se informações para avaliar se os repositórios das ferramentas permitem a recuperação dos elementos dos diagramas a partir das informações salvas.

Os critérios desta categoria correspondem a requisitos funcionais e possuem tipo de resposta 1.

2.7 Critérios baseados na documentação

Os critérios desta categoria demonstram a relação das ferramentas com a respectiva documentação. O objetivo é verificar se existe uma preocupação quanto à documentação e como a mesma é disponibilizada. Foram criados quatro critérios nesta categoria, todos correspondem a requisitos não-funcionais e possuem resposta do tipo 1. A seguir apresenta-se alguns exemplos de critérios desta categoria:

- *Existe help, manuais e documentação que auxilie o usuário a esclarecer dúvidas?* A partir da resposta a este critério é possível verificar a existência de documentação, de que

tipo é a mesma e se realmente auxilia o usuário na utilização da ferramenta.

- *Existe documentação que esclarece dúvidas quando à instalação?* O objetivo é verificar se existe alguma documentação que auxilie a instalação da ferramenta.
- *É possível configurar a forma como a documentação será gerada?* Com este critério é possível verificar se o usuário poderá alterar a documentação gerada pela ferramenta de acordo com suas necessidades.

2.8 Critérios baseados na utilização da ferramenta com relação ao computador utilizado para avaliação

Nesta categoria busca-se informações que permitam avaliar as ferramentas com relação ao computador utilizado para avaliação das mesmas. Para atingir este objetivo foram criados dois critérios:

- *Possui tempo de resposta apropriado?*¹ A partir da resposta a este critério pretende-se detectar como foi a resposta aos processos feitos na ferramenta.
- *Qual o desempenho no computador utilizado?* A intenção é indicar a viabilidade da utilização, orientando o usuário se poderá utilizá-la com a tecnologia que possui ou deverá utilizar mais recursos.

Os critérios desta categoria correspondem a requisitos não-funcionais e possuem resposta descritiva (tipo 2).

3. Aplicação do modelo

A fim de validar o modelo proposto foi definido e elaborado um estudo de caso. Escolheu-se algumas ferramentas e elaborou-se os diagramas da UML com base em um Controle de Simpósio. Inicialmente foram identificados os atores e, logo após, realizou-se um levantamento das responsabilidades de cada um. As responsabilidades foram descritas como pequenas tarefas. A análise destas tarefas intermediárias possibilitou a identificação de prováveis casos de uso que serviram de base para a construção do diagrama correspondente. Em seguida, tornou-se possível a identificação das classes, bem como suas características

¹ este critério também baseou-se na norma ISO/IEC 9126, característica eficiência (sub-característica tempo), mas julgou-se mais apropriado deixá-lo nesta categoria.

e relacionamentos dando origem ao diagrama de classes. A partir da descrição das tarefas intermediárias de cada diagrama de caso de uso foram desenvolvidos os diagramas de seqüência. A seguir, com base na classe Participante, criou-se o diagrama de estados e o diagrama de atividades baseou-se na classe Cancelamento. A Figura 1 apresenta uma parte do Diagrama de Classes construído. Todos os diagramas elaborados encontram-se em [2].

As ferramentas escolhidas para este estudo foram a *Rational Rose C++ Demo*² da IBM/Rational, *PowerDesigner 9.0*³ da Sybase, *Together ControlCenter 6.1*⁴ da Borland, *AllFusion Component Modeler 4.1*⁵ da CA (Computer Associates), *Enterprise Architect 3.51*⁶ da Sparx Systems, *Poseidon for UML Community Edition 1.6*⁷ e *ArgoUML 0.14*⁸. A *Rational Rose* foi escolhida por ser desenvolvida pela mesma empresa da UML; a *PowerDesigner 9.0* por ser uma ferramenta de modelagem muito utilizada no meio acadêmico em geral; a *Poseidon* e a *ArgoUml* por serem ferramentas de código aberto; a *Together ControlCenter*, a *AllFusion Modeler* e a *Enterprise Architect* por serem ferramentas encontradas nas referências utilizadas. Além disso, foi possível encontrar cópias de demonstração de todas as ferramentas acima citadas.

A avaliação das ferramentas foi realizada através da aplicação do estudo de caso nas mesmas e consulta à informações disponibilizadas pelo seu fornecedor por meio da documentação de cada uma delas.

3.1 Resumo dos Resultados

Durante a avaliação foi possível criar a modelagem de acordo com as exigências da UML. Constatou-se que cada ferramenta possui padrões e características diferentes. Os principais resultados constatados com a avaliação são mostrados na Tabela 6.

3.2 Observações finais sobre a avaliação

Através da avaliação verificou-se que cada ferramenta possui características e padrões diferentes que devem ser ressaltados. Identificou-se que em todas as ferramentas é possível criar a modelagem conforme a

UML, porém dependendo da ferramenta isto pode ser feito com maior ou menor dificuldade.

Todas as ferramentas estudadas oferecem várias formas de documentar os projetos, reforçando uma das principais características da UML que é de ser uma linguagem de documentação. Além dos diagramas da UML constatou-se que algumas ferramentas como a *Together ControlCenter*, *Enterprise Architect* e *Allfusion Component Modeler*, oferecem modelagem de negócios, e a *PowerDesigner* possibilita modelagem física e conceitual.

Observou-se, com exceção da versão da *Rational Rose* estudada, que todas as ferramentas preocupam-se com a portabilidade dos seus modelos, mesmo as que não possuem versões para outros sistemas operacionais, pois elas oferecem exportação de seus modelos em XML.

A maioria das ferramentas é bem construída graficamente, a *Poseidon* e a *Together ControlCenter* possuem ícones com os desenhos na forma dos diagramas correspondentes, facilitando a compreensão e agilizando a construção dos mesmos. Além disto, com exceção da *AllFusion Component Modeler*, todos os diagramas podem ser visualizados facilmente através do *browser*, pois são agrupados conforme o tipo de diagrama de acordo com a UML. As empresas que não possuem recursos financeiros ou não desejam gastar para adquirir uma ferramenta podem utilizar a *Argouml* e a *Poseidon*, porque são de código aberto e distribuídas gratuitamente.

4. Conclusões

A UML é uma tendência entre desenvolvedores de aplicações baseadas em modelos orientados a objetos, entretanto, há uma grande dificuldade de encontrar experiências práticas e exemplos mais realistas na literatura, tornando lento o processo de inserção desta metodologia no mercado de trabalho. Existem grandes esforços para mudar este panorama. A prova disto é o investimento do OMG em melhorias na UML lançando constantemente novas versões.

Atualmente, existem várias ferramentas de apoio à UML. A fim de facilitar a avaliação destas ferramentas definiu-se um modelo composto por um conjunto de critérios baseados em requisitos funcionais, não funcionais e normas de qualidade. O objetivo do modelo é auxiliar o usuário para que ele obtenha informações sobre cada ferramenta a fim de que, a partir destas informações possa escolher a que melhor atenda as suas necessidades ou valendo-se dos dados do

² Disponível em: <<http://www.rational.com/>> Acesso 10 abr. 2004.

³ Disponível em: <<http://www.sybase.com/>>. Acesso 10 abr. 2004.

⁴ Disponível em: <<http://www.borland.com/together/controlcenter/>>. Acesso 10 abr. 2004.

⁵ Disponível em: <<http://www3.ca.com/Solutions/Product.asp?ID=1003>>. Acesso 10 abr. 2004.

⁶ Disponível em: <<http://www.sparxsystems.com.au/>>. Acesso 10 abr. 2004.

⁷ Disponível em: <<http://www.gentleware.com/products/index.php3>> Acesso 10 abr. 2004.

⁸ Disponível em: <<http://argouml.tigris.org/>>. Acesso 10 abr. 2004.

estudo, criar uma nova avaliação. Não é objetivo deste modelo indicar qual a melhor ferramenta, bem como abordar custos de aquisição e manutenção.

Sugere-se que após a realização da avaliação, o usuário estabeleça prioridades aos critérios de acordo com suas necessidades. Para os critérios podem ser atribuídos faixas de valores, por exemplo 0 a 5, que demonstrará em que grau a ferramenta atende aquele critério. Assim a ferramenta que melhor atender as necessidades é a que totalizar mais pontos.

Cabe enfatizar que, na seleção de ferramentas, é importante considerar tanto os requisitos funcionais como os não-funcionais. Por esta razão, na elaboração dos critérios do modelo proposto teve-se a preocupação em fazer esta distinção.

O crescente aumento de novas ferramentas disponíveis, incluindo algumas de código aberto e uso livre e a constante atualização das mesmas para versões superiores, motivam a complementação deste trabalho abrangendo mais características das ferramentas avaliadas com a elaboração de novos critérios. Também é possível estender a avaliação realizada para um maior número de ferramentas.

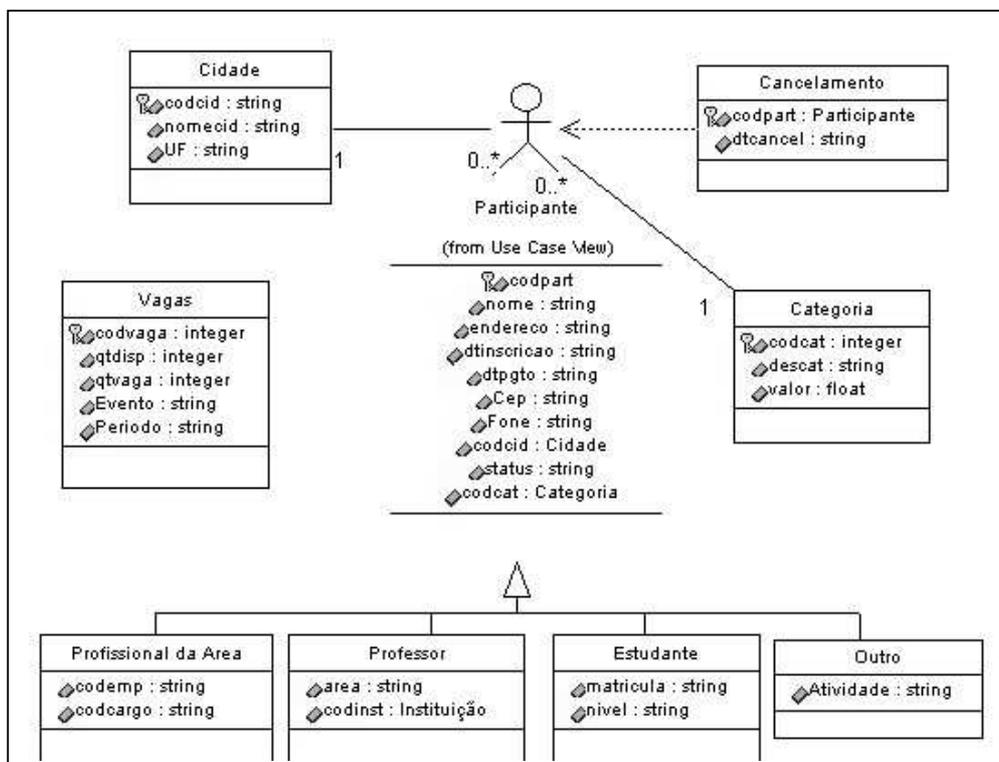


Figura 1 - Parte do Diagrama de Classes

Tabela 6 – Principais Resultados

Item	Resultado
Geração Código e Engenharia Reversa	Todas as ferramentas estudadas possuem estes recursos para pelo menos uma linguagem. Entre as mais utilizadas estão: C, C++, Java.
Documentação Gerada	Todas permitem que se configure a maneira como a documentação será gerada pela ferramenta através da escolha dos tipos de diagramas desejados, elementos e saídas.

Item	Resultado
Importação e Exportação de Arquivos	Com exceção da <i>Rational</i> , todas exportam dados em XML. A <i>PowerDesigner</i> importa os modelos criados na <i>Rational</i> , mas o diagrama de classes fica com os objetos e ligações confusos na ferramenta, após a importação. A <i>Poseidon</i> abre modelos criados na <i>Argouml</i> e vice-versa.
Portabilidade	A <i>Enterprise Architect</i> e <i>Allfusion</i> possuem versões apenas para o <i>Windows</i> , as demais (exceto a <i>PowerDesigner</i> que não identificou-se) contêm versões para outros sistemas operacionais. A <i>Poseidon</i> e a <i>Argouml</i> independem do mesmo.
Hardware e Software Necessários	128 MB de RAM recomendada pelo fornecedor é insuficiente para utilização da <i>Poseidon</i> e da <i>Argouml</i> . Para a <i>Together</i> é recomendando no mínimo 512MB. As ferramentas <i>Together</i> e <i>Allfusion</i> requerem grande espaço em disco (a partir de 100MB); para <i>Enterprise Architect</i> , <i>Argouml</i> e <i>Rational Rose</i> é necessário até 15MB e para a <i>Poseidon</i> e <i>PowerDesigner</i> não foi possível identificar, porém o tamanho do arquivo de instalação das mesmas é superior a 60MB. O <i>drive</i> de CD só é necessário para instalação das ferramentas. Recomenda-se uma placa de vídeo com resolução mínima de 800x600, exceto para a <i>Together</i> pois o fornecedor recomenda 1024x768.
Repositório de Dados	Somente a <i>Allfusion</i> necessita de instalação de uma base de dados específica (<i>MS SQL Server</i>). A <i>PowerDesigner</i> , <i>Enterprise Architect</i> e <i>AllFusion Component Modeler</i> possuem repositórios de dados que permitem controlar usuários ou grupos de usuários, atribuindo-lhes permissões para os projetos; a <i>Together</i> também contém este controle, porém como acontece com as demais, o repositório de dados é utilizado apenas como um browser para visualização dos elementos dos diagramas. Com exceção da <i>Poseidon</i> e da <i>Rational Rose</i> (que não foi possível identificar), todas têm suporte a algum banco de dados.
Interface	Com exceção da <i>AllFusion Component Modeler</i> , todas são muito bem organizadas graficamente.
Documentação	Todas são acompanhadas de documentação, sendo também disponibilizadas na página correspondente.
Conhecimentos Prévios	Com exceção da <i>AllFusion Component Modeler</i> , não houve dificuldades na manipulação das ferramentas.
Relação com a UML	Com exceção da <i>Rational Rose</i> , é possível criar um diagrama de classes com herança múltipla. Na <i>Poseidon</i> , <i>Together</i> e <i>Enterprise Architect</i> é possível criar todos os diagramas da UML. Na <i>AllFusion</i> não foi possível construir o diagrama de casos de uso. A <i>PowerDesigner</i> possibilita a criação dos diagramas de atividades, classes, casos de uso, componentes e seqüência; na <i>Rational</i> não é possível construir o diagrama de atividade e na <i>Argouml</i> o de seqüência. Todas possibilitam a utilização de estereótipos, elas já vêm com alguns pré-definidos e outros podem ser criados.
Fornecedor	Os fabricantes já estão há muito tempo no mercado, exceto a <i>Gentleware</i> (2000). Alguns se uniram, como a IBM e a <i>Rational</i> e a <i>Computer Associates</i> à <i>Platinum Software</i> (fabricante da <i>Paradigm Plus</i> , atual <i>AllFusion Component Modeler</i>). Todas as ferramentas estudadas possuem cópia de demonstração.

5. Referências Bibliográficas

- [1] Booch, G. et al (2000) "UML: Guia do Usuário". Rio de Janeiro: Campus, 2000.
- [2] Foresti, J. (2003) "Avaliação de Ferramentas de Apoio à UML", Monografia, Ciência da Computação, Universidade de Passo Fundo.
- [3] Furlan, J.D. (1998) "Modelagem de Objetos através da UML". São Paulo: *Makron Books*.
- [4] Maldonado, J.C. et al (2001) *Qualidade de Software: Teoria e Prática*. São Paulo: *Prentice Hall*.
- [5] Matos, A.V. (2002) "UML: Prático e Descomplicado". São Paulo: Érica, 2002.
- [6] Medeiros, E. (2004) *Desenvolvendo Software com UML 2.0*. São Paulo: *Pearson Makron Books*.
- [7] *Objects By Design*. Disponível em http://www.objectsbydesign.com/tools/modeling_tools.html. Acesso em 28 ago. 2004.