



**Universidade Federal de Pernambuco**  
**Centro de Informática**  
**Bacharelado em Ciência da Computação**



**Carlos Roberto da Silva Júnior**  
(crsj@cin.ufpe.br)

## **Reestruturação e Expansão do Methodology Explorer**

---

**Trabalho de Graduação em Ciência da  
Computação do Centro de Informática da  
Universidade Federal de Pernambuco**

**Orientador: Hermano Perrelli de Moura**  
(hermano@cin.ufpe.br)

Recife, 12 de março de 2003

# Agradecimentos

Em primeiro lugar, agradeço a Deus acima de tudo. Ele que é fonte inspiradora e responsável maior pelo sucesso de todos os homens na Terra. Sem a sabedoria, a fortaleza e a coragem vindos de Deus, o sucesso não teria sido alcançado.

Gostaria de agradecer de todo coração a “painho” e “mainha” que me apoiaram desde meus primeiros passos no maternal até o presente momento. Muito obrigado, meu pai, por aquela surra quando faltei aula na alfabetização. Hoje estamos vendo os resultados que elas produziram. Obrigado minha mãe, por aquelas lágrimas no dia da surra. Se antes eram lágrimas de piedade, agora são lágrimas de alegria.

Agradeço a minha irmã, Cláudia, por ter brincado comigo nos tempos em que a vida não tinha grandes preocupações nem responsabilidades. Toda vez que precisei trocar o dia pela noite, me lembrava do quanto aquele tempo faz falta.

Este trabalho, dedico especialmente a você, Sandra Paula. A baixinha mais “chata” e mais maravilhosa do mundo, por quem me apaixonei. Sem você não teria inspiração e forças para continuar vencendo obstáculos.

Também de agradeço ao professor Fernando Fonseca, meu primeiro e eterno orientador. Seu apoio, sua confiança e, principalmente, suas orientações no SEI, tenham sido elas nas críticas ou nos elogios, me engrandeceram profundamente.

Não posso deixar também de agradecer aos “pais” do meu trabalho de graduação: Suzana e Hermano. Suzana, você foi a minha orientadora de consideração neste projeto. Suas “orientações” quando comecei a ter contato com o material produzido me economizaram alguns cabelos brancos. Hermano, pela pessoa que és, foi muito tê-lo como orientador. Aliás, agora os adjetivos aumentando: amigo, chefe, orientador...

Por fim, agradeço a todos os meus familiares (avós, tios, primos, ...) e a todos aqueles que estiveram comigo durante toda minha vida acadêmica, desde o Colégio de São Bento (aqueles meus amigos do terceiro ano são demais) até meus novos amigos de faculdade, em especial meu “grupo de projeto” (Mardoqueu, Renato e Baiano) e meu “grupo de trabalho” (Clóvis e Júlio).

A todos vocês, meus sinceros agradecimentos!

# Sumário

<b>LI STA DE FIGURAS.....</b>	<b>IV</b>
<b>RESUMO.....</b>	<b>VI</b>
<b>ABSTRACT.....</b>	<b>VI</b>
<b>1 INTRODUÇÃO .....</b>	<b>1</b>
<b>2 MOTIVAÇÃO PARA O METHODOLOGY EXPLORER .....</b>	<b>3</b>
<b>3 ANTECEDENTES E A VERSÃO ALFA .....</b>	<b>5</b>
3.1 A INTERFACE GRÁFICA DA VERSÃO ALFA.....	5
3.2 FUNCIONALIDADES PRESENTES .....	6
3.2.1 Criação e Remoção de Componentes.....	7
3.2.2 Visualização e Alteração de Componentes.....	7
3.2.3 Composição de Componentes.....	7
<b>4 ANÁLISE CRÍTICA DA VERSÃO ALFA: O RECOMEÇO .....</b>	<b>8</b>
4.1 ANÁLISE DE USABILIDADE.....	8
4.1.1 Participantes .....	8
4.1.2 Procedimento .....	9
4.1.3 Análise dos Dados.....	10
4.1.4 Resultado da Avaliação e Problemas Identificados.....	11
4.1.4.1 Processamento Lento .....	11
4.1.4.2 Pouca Visibilidade das Funcionalidades .....	12
4.1.4.3 Ausência de Atalhos.....	12
4.1.4.4 Ausência de <i>Feedback</i> para o Usuário.....	12
4.1.4.5 Conceitos Distintos Representados numa Mesma Tela .....	12
4.2 ANÁLISE DE PROJETO E IMPLEMENTAÇÃO.....	13
4.2.1 Dependência com o RUP.....	13
4.2.2 Acessos ao Banco de Dados .....	14
<b>5 MELHORIAS E EVOLUÇÕES DE PROJETO .....</b>	<b>15</b>
5.1 A REMODELAGEM ARQUITETURAL .....	15
5.1.1 Elementos Básicos do Methodology Explorer 1.0.....	16

5.1.1.1	Componente .....	16
5.1.1.2	Tipo de Componente .....	16
5.1.1.3	Metodologia .....	17
5.1.1.4	Conteúdo Anexo e Regras de Coleção.....	17
5.1.2	<i>Modelagem dos Dados</i> .....	18
5.2	ACESSO CONTROLADO À BASE DE DADOS .....	18
<b>6</b>	<b>O NOVO DESIGN DA INTERFACE GRÁFICA.....</b>	<b>20</b>
6.1	ATALHOS E A DIVERSIFICAÇÃO DE OPÇÕES PARA REALIZAÇÃO DE TAREFAS .....	21
6.1.1	<i>Barra de Ferramentas</i> .....	21
6.1.2	<i>Menus de Pop-up</i> .....	21
6.1.3	<i>Teclas de Combinação</i> .....	22
6.2	SEPARAÇÃO LÓGICA DE CONCEITOS .....	22
6.2.1	<i>Metodologia de Trabalho</i> .....	23
6.2.2	<i>Divisão da Tela em Painéis Conceituais</i> .....	23
6.2.3	<i>Conceito de Abertura e Fechamento de Metodologias</i> .....	24
6.3	MAPEAMENTO DE FUNCIONALIDADES EM CONTROLES GRÁFICOS .....	25
6.3.1	<i>Uso de Ícones</i> .....	25
6.3.2	<i>Criação de Novos Controles</i> .....	27
<b>7</b>	<b>OS NOVOS RECURSOS E FUNCIONALIDADES.....</b>	<b>28</b>
7.1	ANEXAÇÃO DE ARQUIVOS.....	29
7.2	IMPORTAÇÃO E EXPORTAÇÃO DE METODOLOGIAS .....	30
7.3	METHODOLOGY EXPLORER INSTANCE CREATOR .....	32
7.4	METHODOLOGY EXPLORER ANALYSER.....	34
7.5	METHODOLOGY EXPLORER WEB PUBLISHER .....	36
<b>8</b>	<b>CONCLUSÕES .....</b>	<b>39</b>
8.1	RESULTADOS ALCANÇADOS .....	39
8.2	DIFICULDADES ENCONTRADAS.....	40
8.3	TRABALHOS FUTUROS .....	41
	<b>REFERÊNCIAS .....</b>	<b>43</b>

# Lista de Figuras

FIGURA 1. TELA PRINCIPAL DA VERSÃO ALFA DO METHODOLOGY EXPLORER....	6
FIGURA 2. O SOFTWARE QSR NUD*IST, COM AS CATEGORIAS DE INFORMAÇÕES EXTRAÍDAS E ORGANIZADAS NUMA ESTRUTURA EM ÁRVORE. ....	11
FIGURA 3. O MODELO DE DADOS LÓGICO DA VERSÃO ALFA .....	13
FIGURA 4. O NOVO MODELO DE DADOS DA VERSÃO METHODOLOGY EXPLORER 1.0 .....	18
FIGURA 5. A TELA INICIAL DO METHODOLOGY EXPLORER 1.0 .....	20
FIGURA 6. BARRA DE FERRAMENTAS DO METHODOLOGY EXPLORER .....	21
FIGURA 7. USO DE UM MENU DE <i>POP-UP</i> .....	22
FIGURA 8. TELA DIVIDIDA CONCEITUALMENTE: METODOLOGIA DE TRABALHO E VISÃO DETALHADA .....	23
FIGURA 9. A VISÃO DAS METODOLOGIAS .....	24
FIGURA 10. ABERTURA DE UMA METODOLOGIA.....	25
FIGURA 11. AS ÁRVORES DE METODOLOGIAS COM NÓS ICONIZADOS.....	26
FIGURA 12. ESCOLHA DO TIPO DE CONTEÚDO NA CRIAÇÃO DE UM COMPONENTE .....	29
FIGURA 13. INSERÇÃO DE CONTEÚDO TEXTUAL .....	30
FIGURA 14. ITEM DE MENU PARA EXPORTAÇÃO DE METODOLOGIA.....	31
FIGURA 15. METODOLOGIA SENDO EXPORTADA.....	31
FIGURA 16. A METODOLOGIA EXPORTADA NO SISTEMA DE ARQUIVOS.....	32
FIGURA 17. <i>METHODOLOGY EXPLORER INSTANCE CREATOR</i> .....	33
FIGURA 18. ALGUNS PASSOS DO <i>INSTANCE CREATOR</i> .....	33

FIGURA 19. NOÇÃO DE INSTÂNCIA: B É UMA INSTÂNCIA DE A.....	34
FIGURA 20. NOÇÃO DE VARIANTE: B É UMA VARIANTE DE A .....	34
FIGURA 21. A É INSTÂNCIA E VARIANTE DE B E VICE-VERSA .....	35
FIGURA 22. <i>METHODOLOGY EXPLORER ANALYSER</i> .....	35
FIGURA 23. PASSOS DO <i>ANALYSER</i> .....	36
FIGURA 24. <i>METHODOLOGY EXPLORER WEB PUBLISHER</i> .....	37
FIGURA 25. PASSOS DO <i>WEB PUBLISHER</i> .....	37
FIGURA 26. EXEMPLO DE <i>SITE</i> GERADO PELO <i>METHODOLOGY EXPLORER WEB PUBLISHER</i> .....	38

# Resumo

O Methodology Explorer foi idealizado como uma ferramenta voltada para definição de metodologias, baseada em criação e reutilização de componentes. Sua existência ajuda o trabalho de projetistas de metodologias e engenheiros de processos e vem a preencher uma carência na área de processos de software.

Este trabalho teve como objetivos avaliar a idéia Methodology Explorer e o estado em que se encontrava seu primeiro protótipo para, a partir de daí, identificar aspectos negativos e prover uma completa reestruturação do sistema. Além disso, foi um dos objetivos principais também evoluir a ferramenta através da criação de novos e poderosos recursos. Recursos esses considerados essenciais para o sucesso da ferramenta.

**Palavras Chave:** engenharia de software, processos de software, metodologias, reuso

# Abstract

Methodology Explorer was realized as tool directed toward to definition of methodologies, based in creation and reuse of components. Its existence helps methodologies designers and processes engineers work and come to fill a lack at software processes area.

This work had the objectives to evaluate the Methodology Explorer idea and the state its first prototype was for, from there, to identify negative aspects and provide a complete system reorganization. Moreover, one of the major objectives was to evolve the tool too through the creation of new and powerful resources. These resources are considered essential to tool success.

**Keywords:** software engineering, software processes, methodologies, reuse.

# 1 Introdução

Este trabalho tem como objetivo estudar a implementação da ferramenta Methodology Explorer, cujo estado já se encontrava numa fase de desenvolvimento primordial, e a partir daí, propor e executar uma série de modificações necessárias para melhorar a ferramenta e expandir suas funcionalidades.

O presente documento descreve o projeto inicial e seus objetivos e relata o estado da ferramenta quando iniciado o trabalho de reestruturação e expansão. Segue-se a isso, toda a avaliação dessa versão e as deficiências encontradas. Posteriormente, são detalhados todo o trabalho realizado na remodelagem da ferramenta, os novos recursos criados e os resultados alcançados.

O documento está dividido em seções definidas pelos capítulos seguintes. O Capítulo 2 apresenta o projeto Methodology Explorer, seus objetivos e motivações. Neste capítulo são vistos o público alvo da ferramenta, suas aplicações na engenharia de software e sua importância no dia-a-dia dos usuários.

O Capítulo 3 traz um breve histórico da ferramenta até antes do início deste trabalho. São apresentados detalhes da origem do projeto, seus idealizadores e produtos gerados.

No Capítulo 4 é visto o que podemos chamar de início deste trabalho. Neste capítulo são apresentados os procedimentos utilizados para avaliação do estado em que se encontrava a ferramenta e são identificados os aspectos mais críticos da versão inicial. Estes serviram como parâmetros para o planejamento da nova versão da ferramenta e para a execução das alterações e expansões.

No Capítulo 5 são abordadas as principais modificações realizadas na estrutura interna da ferramenta, que compreendeu tanto a remodelagem das classes como também do novo projeto da base de dados. Além das modificações arquiteturais, este capítulo também aborda algumas mudanças de paradigmas utilizados, como o das atualizações retardadas, que postergam as atualizações no banco de dados, antes feitas instantaneamente.

No Capítulo 6 podemos ver o novo projeto da interface gráfica, que se baseou nos resultados avaliados no capítulo 4 para produzir uma interface mais amigável, com elementos melhor distribuídos, mais navegável e fácil de utilizar. As técnicas utilizadas na criação desta interface também são detalhadas neste capítulo.



O Capítulo 7 é dedicado a detalhes sobre algumas das principais funcionalidades criadas para expandir os recursos oferecidos pelo Methodology Explorer. É neste capítulo que são tratadas as funcionalidades totalmente novas desta versão. Algumas delas, inclusive, foram criadas através da implementação de ferramentas, no estilo de *wizards*, que foram incorporadas ao Methodology Explorer.

Finalmente, o Capítulo 8 conclui este trabalho com os resultados alcançados, as dificuldades encontradas na execução do projeto e possíveis trabalhos futuros já identificados.

## 2 Motivação para o Methodology Explorer

No mundo da engenharia de software, percebe-se uma grande carência de ferramentas direcionadas às atividades de definição de metodologias para o processo de desenvolvimento de software. Diante dessa carência, o trabalho dos profissionais da área tem sido dificultado, pois, para cada novo cliente, a definição de sua metodologia requer que uma série de elementos de documentação precise ser refeita. Isso inclui *templates*, artefatos, *sites*, figuras, planilhas, etc.

Mesmo quando as metodologias de dois clientes diferentes são bem parecidas, o retrabalho é bastante alto. Isso ocorre em decorrência da grande quantidade de trabalho “manual” (não automático) que o engenheiro de processos precisa realizar. Por exemplo, para reutilizar um fluxo de atividades de uma metodologia A em uma metodologia B, é preciso, primeiramente, saber que existe tal fluxo. Além disso, é necessário localizar os documentos referentes a esse fluxo no ambiente de trabalho da empresa. Vencidos os dois primeiros obstáculos, é necessário ainda analisar os conteúdos dos documentos para validar a real adequação dos mesmos à metodologia que está sendo desenvolvida.

Esse exemplo demonstra claramente algumas dificuldades presentes no dia-a-dia de um engenheiro de processos. Uma observação relevante no exemplo dado é que as tarefas listadas são todas tarefas que não exigem do profissional conhecimento técnico muito apurado. As atividades técnicas do profissional (definir componentes reusáveis, definir processos, elaborar documentos, entre outras) acabam dividindo espaços com atividades puramente burocráticas.

Um ganho enorme de produtividade poderia ser alcançado se as atividades não-intelectuais pudessem ser automatizadas. Melhor dizendo, o trabalho do profissional de engenharia de software seria bem melhor realizado com o uso de ferramentas que ajudassem a acelerar as atividades inclusas no processo de definição de metodologias. Uma ferramenta específica para realização de tais atividades poderia facilitar a localização de documentos, como também fornecer uma visualização maior dos componentes presentes nas metodologias já desenvolvidas ou em desenvolvimento pela equipe de profissionais de uma empresa.

Melhor do que simplesmente um ambiente para definição de metodologias, seria um ambiente que já contivesse metodologias genéricas para empresas de software e que também permitisse gerar e armazenar as próprias instâncias dessas metodologias.

Assim, o trabalho de definir uma nova metodologia tornar-se-ia o trabalho de instanciar uma metodologia genérica ou de compô-la a partir de outras já definidas neste ambiente.

O projeto Methodology Explorer [1] foi idealizado com objetivo de produzir uma ferramenta voltada para o processo de desenvolvimento de software. Esta ferramenta deveria facilitar o trabalho dos engenheiros de processos e projetistas de metodologias através da criação de metodologias baseadas em componentes. Componentes são elementos mais ou menos independentes que compõem uma metodologia e podem ser manipulados isoladamente. Uma metodologia seria composta por conjuntos de componentes e estes também compostos por conjuntos de sub-componentes. Os componentes poderiam ser definidos pelo usuário e/ou reusados a partir de outras metodologias já criadas com o uso da ferramenta.

A ferramenta, que recebeu o nome do projeto, Methodology Explorer, também deveria conter algumas *features* que possibilitassem a publicação da metodologia. No trabalho de definição de uma metodologia para um determinado cliente, o projetista deve entregar, como seu produto, um conjunto de documentos em algum formato determinado. É muito comum que a metodologia seja entregue como um *site* HTML contendo *links* para páginas HTML e para outros documentos em diferentes formatos. Levando em consideração essa necessidade, o Methodology Explorer se torna muito mais útil com as funcionalidades de publicação de metodologias em formatos como HTML.

Assim o Methodology Explorer foi concebido como uma ferramenta para definição de metodologias, permitindo reuso de componentes e publicação da metodologia juntamente com todos os seus componentes. Sua utilização estimula a criação de componentes padronizados, facilita o trabalho dos engenheiros de processos, além de melhorar a produtividade desses profissionais e a qualidade do processo de software.

## 3 Antecedentes e a Versão Alfa

A necessidade de uma ferramenta como o Methodology Explorer é antiga, tanto que há alguns anos a idéia do projeto já era discutida no Centro de Informática da UFPE. Em agosto de 2001, a aluna Suzana Mesquita de Borba Maranhão iniciou os primeiros passos do Methodology Explorer com a criação de um protótipo básico [2], como resultado de sua pesquisa de iniciação científica. Suzana contou com a orientação professor Hermano Perrelli de Moura, idealizador de todo o projeto [1].

O protótipo, que de agora em diante passaremos a chamá-lo de Versão Alfa, foi desenvolvido em Java e tentou implementar algumas das funcionalidades básicas pretendidas para a ferramenta. Nesta primeira versão, já era possível a definição de metodologias baseadas no *Rational Unified Process* [3], com reuso de componentes e armazenamento dos componentes no banco de dados da ferramenta.

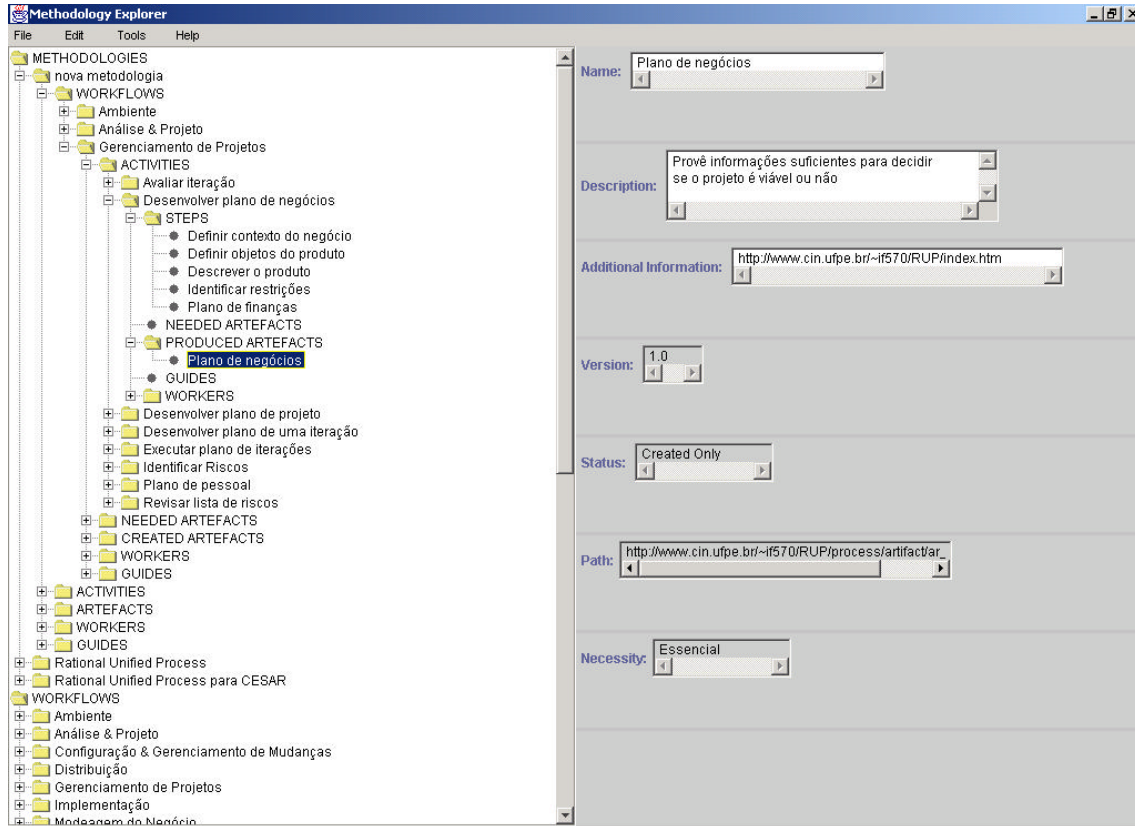
### 3.1 A Interface Gráfica da Versão Alfa

A Versão Alfa do Methodology Explorer permitia que usuário criasse metodologias, a partir da criação de seus componentes. Dentro da ferramenta, o projetista enxergava uma metodologia como uma estrutura hierárquica composta por nós e cada nó correspondia a um componente específico.

Todas as metodologias criadas com a ferramenta eram baseadas no RUP (*Rational Unified Process*). Hierarquicamente, o RUP pode ser assim representado de forma simplificada:

- Uma metodologia possui N disciplinas (ou fluxo);
- Uma disciplina (ou fluxo) possui N atividades;
- Uma atividade possui N passos e artefatos de entrada e saída;
- Cada atividade é desempenhada por um determinado papel.

Desse modo, os componentes da metodologia RUP, representados no Methodology Explorer, seriam metodologia, disciplinas, atividades, passos, papéis, artefatos, etc. Assim, cada elemento do RUP correspondia a nós da árvore de componentes. A Figura 1 apresenta a tela principal do protótipo, onde se pode observar claramente os nós.



**Figura 1.** Tela principal da Versão Alfa do Methodology Explorer

A tela acima é dividida em três partes:

- Na parte superior se encontra a **barra de menus**, a partir da qual se obtém acesso às principais funcionalidades do sistema;
- Do lado esquerdo se encontra o **painel de navegação**. Este apresenta todos os componentes de metodologias organizados numa estrutura em árvore. As raízes de cada árvore correspondem aos nomes dos tipos de componentes existentes no RUP (fluxos, artefatos, atividades, papéis). Abaixo das raízes aparecia cada um dos componentes em si (os vários fluxos, artefatos, etc). Cada nó da árvore é um componente ou nome de um tipo de componente da metodologia.
- Do lado direito se encontra o **painel de detalhes**, neste painel são apresentadas informações sobre algum componente selecionado no painel de navegação.

### 3.2 Funcionalidades Presentes

Esta primeira versão oferecia algumas funcionalidades, dentre aquelas mencionadas no capítulo anterior.

### 3.2.1 Criação e Remoção de Componentes

Na barra de menus, através do menu *File*, era possível criar e remover componentes de metodologias de qualquer um dos tipos de elementos presentes no RUP. Uma vez criados esses componentes, eles eram adicionados, como nós, em algumas das árvores de componentes do painel de navegação.

### 3.2.2 Visualização e Alteração de Componentes

Um simples clique em algum dos componentes do painel de navegação permitia que os detalhes do mesmo fossem visualizados no painel de detalhes. Neste painel, informações, como nome do componente, descrição, versão e status, poderiam tanto ser visualizadas como também alteradas livremente.

### 3.2.3 Composição de Componentes

A idéia de reuso de componentes, tão desejada no projeto, estava presente neste protótipo através da composição de novas metodologias por componentes já criados em outras metodologias. Essa funcionalidade era realizada com o uso da idéia de “*copy and paste*” (copiar e colar), muito utilizada em outros *softwares*.

O menu *Edit* possuía os seus itens *copy* e *paste*. Para compor uma metodologia, bastava selecionar o componente que deveria ser reutilizado, copiá-lo através do menu, selecionar o local onde ele deveria ser colado e confirmar este passo no item de menu *paste*. Assim o componente, juntamente com todos os nós abaixo dele, era copiado na hierarquia de uma outra metodologia.

## 4 Análise Crítica da Versão Alfa: o Recomeço

Após o trabalho de Suzana [2], que resultou na criação da Versão Alfa, o Methodology Explorer passou a ser divulgado em palestras [4] e artigos [5]. Foi a partir daí que o autor do presente trabalho manifestou interesse em dar continuidade ao desenvolvimento da ferramenta. O objetivo naquele momento seria aprimorar o protótipo e implementar mais um novo conjunto de funcionalidades essenciais para que, ao fim dos trabalhos, a versão 1.0 do Methodology Explorer pudesse ser produzida.

A nova versão deveria partir daquilo que já existia da Versão Alfa. Assim, o primeiro desafio deste trabalho foi procurar documentos, código fonte e a própria autora do protótipo para tomar mais contato com a idéia do projeto e com aquilo já realizado. Além dessas informações, que permitiriam a identificação de funcionalidades ainda não produzidas, se fazia necessário analisar de forma crítica aquelas também já criadas. Em outras palavras, como o que havia sido produzido era apenas um protótipo, seria importante analisar aspectos de projeto, implementação e usabilidade da ferramenta para que as principais deficiências pudessem ser eliminadas. A idéia seria estudar de forma cautelosa a ferramenta e prover a reestruturação daquilo que fosse necessário antes de partir para expansão de seus recursos.

### 4.1 Análise de Usabilidade

A usabilidade do protótipo foi avaliada com realização de um procedimento sistemático que utilizava métodos empíricos [6] para ajudar a evidenciar prováveis problemas de usabilidade na de interface. Esse método contava com a participação ativa de usuários no processo. Todo o procedimento de análise [7], juntamente com os resultados obtidos, será descrito nas próximas seções.

#### 4.1.1 Participantes

A avaliação da usabilidade deste protótipo contou com a participação voluntária de estudantes de graduação em ciência da computação representando o papel de usuários finais da ferramenta. Foram selecionados estudantes que já possuíam alguma experiência com os conceitos de engenharia e processos de software. Para diferenciar os perfis dos participantes, foram escolhidos usuários “experientes” e “iniciantes”, ou seja, participantes com diferentes graus de experiência prática.

Os experientes seriam aqueles que já trabalhavam com engenharia de software e que, portanto, teriam maior facilidade de adaptação ao Methodology Explorer. O perfil de iniciante foi preenchido com estudantes que tinham conhecimento técnico-teórico sobre o assunto, mais que não possuíam nenhuma experiência profissional prática com o uso de processos de software.

#### **4.1.2 Procedimento**

Inicialmente, cada participante foi submetido a uma breve explanação sobre o funcionamento da ferramenta, seus objetivos e aplicações e, em seguida foi feita uma breve demonstração de uso. Nessa demonstração, o tutor apresentou a área de trabalho inicial e cada elemento que compõe a ferramenta e simulou o trabalho de um usuário enquanto utilizando a ferramenta. A demonstração visava consolidar, a partir de exemplos práticos na ferramenta, as explicações que foram dadas na explanação verbal inicial. Durante essas primeiras atividades os participantes também podiam tirar dúvidas sobre os conceitos e funcionalidades da ferramenta.

Após essa sessão, cada participante, separadamente, deveria ser submetido a sessões de interação direta com a ferramenta. Essa interação buscava identificar as dificuldades de cada um na realização das tarefas para poder evidenciar falhas de usabilidade da interface. As sessões consistiam na realização, por parte de cada participante, de um conjunto de atividades definidas pelo tutor.

Durante a sua interação com a ferramenta, o usuário deveria utilizar a técnica empírica do **protocolo verbal** [6]. Essa técnica consistia em explicitar oralmente tudo aquilo que fazia ou pensava durante a realização das tarefas. Os usuários foram instruídos a falar constantemente sobre tudo o que eles estavam fazendo, pensando em fazer, e até mesmo sobre o que sentiam ao usar a ferramenta e sobre as respostas da mesma. Dessa forma todas as impressões dos participantes eram explicitadas: se eles estavam encontrando facilmente as funcionalidades, se estavam gostando dos resultados, se a ferramenta lhes irritava por algum motivo, etc. Resumindo, o protocolo verbal era utilizado para que o participante externasse todo o seu sentimento e dificuldades durante a realização das tarefas. Isso fornecia uma grande quantidade de informações a partir das quais evidências sobre os problemas da interface puderam ser extraídas.

Os participantes realizaram cada tarefa praticando o protocolo verbal de forma intensiva. Toda a interação foi gravada para que os comentários pudessem ser avaliados posteriormente. Após realização das atividades, foi pedido que cada usuário fizesse comentários sobre a ferramenta e sobre sua interação em si, além de sugerir novas



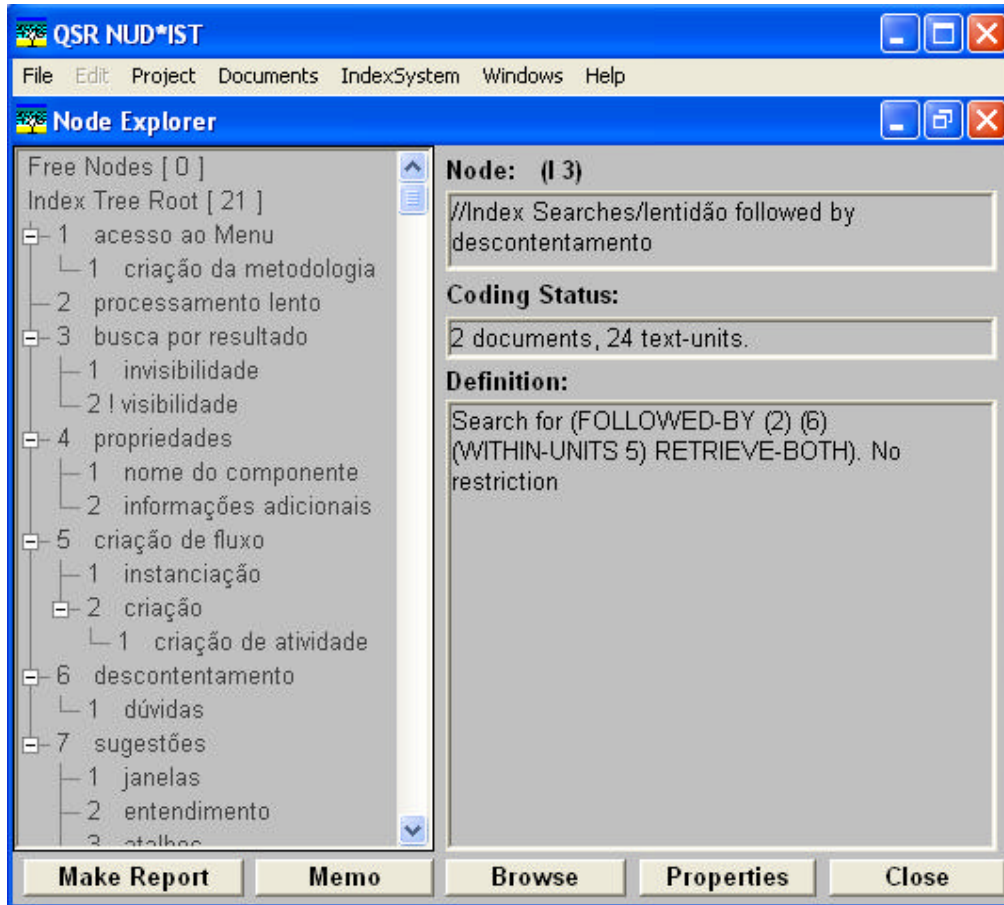
funcionalidades ou possíveis melhoramentos. Tudo isso também foi gravado fornecendo uma fonte maior de informações para a análise da interface.

Quanto ao protocolo verbal cabe aqui algumas observações. Essa é uma técnica que não é facilmente realizada por qualquer participante, pois envolve lidar com a parte psicológica dos participantes. Algumas pessoas mais inibidas têm dificuldades para expressar aquilo que estão sentindo, já outras mais desinibidas se sentem mais à vontade com o protocolo. Era importante que cada participante se envolvesse o máximo com a técnica e que as diferenças de personalidade fossem minimizadas. Para que isso ocorresse, todos os participantes foram submetidos a uma bateria de treinos para se adaptarem ao protocolo verbal. Só após bem treinados, eles realizaram as sessões de interação com a interface do Methodology Explorer.

#### **4.1.3 Análise dos Dados**

Depois de gravadas todas as interações, foi feita a análise dos dados. A intenção era encontrar evidências de problemas de usabilidade através da comparação entre o desempenho dos participantes. As fontes dados para análise foram as gravações do protocolo verbal utilizado durante a realização das atividades e os comentários dos usuários após sua finalização.

Comportamentos repetidos ou sentimentos de frustração comuns aos participantes em considerados evidências de possíveis problemas de usabilidade. Essas informações puderam ser facilmente extraídas com o uso de um ferramental adequado para tal tipo de análise. A ferramenta utilizada para a análise foi o QSR NUD\*IST [10]. Essa ferramenta recebia como entrada a transcrição dos protocolos verbais gravados nas interações e, a partir daí, o texto poderia ser trabalhado a partir de marcações, categorizações, busca e cruzamentos de informações. Dessa forma, as evidências puderam ser facilmente encontradas. A Figura 2, mostra a tela principal da ferramenta com uma parte da árvore de informações obtidas a partir da análise, no seu lado esquerdo.



**Figura 2.** O software QSR NUD\*IST, com as categorias de informações extraídas e organizadas numa estrutura em árvore.

#### 4.1.4 Resultado da Avaliação e Problemas Identificados

A análise realizada permitiu que uma série conclusões pudesse ser extraída sobre o estado atual de usabilidade do protótipo do Methodology Explorer. A seguir são listados os problemas mais críticos identificados.

##### 4.1.4.1 Processamento Lento

Para cada atividade realizada pelo usuário, o sistema consumia um tempo considerável para processamento das informações e apresentação dos resultados. Exemplos disso ocorreram quando o usuário tentava criar ou alterar um componente. O sistema levava cerca de quatro segundos para atualizar toda a hierarquia de componentes no painel de navegação. Esse fato ultrapassava o limite de tolerância de todos os usuários, fazendo-os reclamar constantemente da demora no processamento.

#### 4.1.4.2 Pouca Visibilidade das Funcionalidades

Os usuários não reconheciam de imediato o passo-a-passo a ser feito para realização das atividades. As funcionalidades estavam, quase que em sua maioria, restritas ao acesso através do menu. Muitas vezes, os usuários ficavam a procurar as funcionalidades sem saber onde encontrá-las.

#### 4.1.4.3 Ausência de Atalhos

Não existia nenhuma forma para encurtar os passos de atividades repetitivas. Todas as atividades se concentravam basicamente nos menus, sem teclas de atalho ou barra de ferramentas para as funcionalidades mais acessadas. Isso contribuía fortemente para que as atividades fossem bastante demoradas. Um exemplo bem claro é a atividade de copiar e colar um componente, que necessitava de dois acessos ao menu. Primeiro o usuário precisa selecionar um componente, ir até o menu para clicar na opção *copy* e depois ainda precisaria clicar no local onde o objeto seria incluído para ainda ter que voltar ao menu e escolher a opção *paste*.

#### 4.1.4.4 Ausência de *Feedback* para o Usuário

O processamento lento de algumas funções gerava novos problemas. Durante o processamento dos dados, o usuário não era informado se a operação estava sendo realizada ou se havia terminado. O usuário precisava estar atento a interface para perceber se os resultados de suas ações tiveram algum efeito ou não, pois nada lhe era informado.

#### 4.1.4.5 Conceitos Distintos Representados numa Mesma Tela

Um outro ponto bastante criticado foi a grande quantidade de informações diferentes que eram mostradas na mesma tela. A tela de navegação, em especial, apresentava todos os componentes já existentes na ferramenta juntamente com os componentes que o usuário estava trabalhando. Era impossível identificar o que já existia e o que era novo. Não existia nada que diferenciasse uma árvore de componentes de outra. Tudo ficava misturado na mesma janela.

Outra mistura de conceitos era a não diferenciação entre uma criação de metodologia (criação desde o início) e uma instanciação de metodologia (criação a partir de reuso de componentes). Dentro da ferramenta, não existiam controles (botões, *wizards*) que mapeassem diretamente nesses recursos.

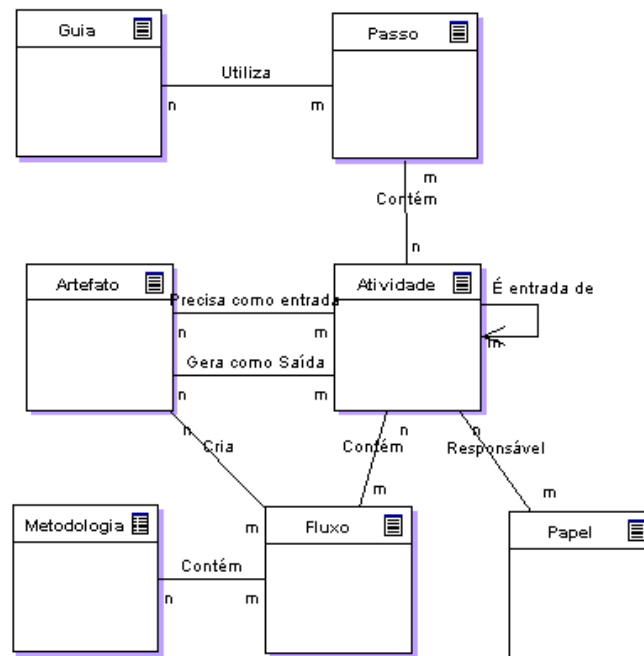
## 4.2 Análise de Projeto e Implementação

Diferentemente da análise de usabilidade, a análise do projeto e da implementação não contou com um experimento sistemático realizado com usuários, até mesmo por não se aplicar a este caso. Este consistiu em analisar minuciosamente documentação e código fonte produzidos, a fim de validar se a arquitetura e a implementação satisfaziam os requisitos pretendidos e de identificar aspectos que carecessem de reestruturações.

Como resultados, foram identificados dois itens que poderiam ser melhorados na arquitetura: a dependência com uma metodologia específica, o RUP, e a quantidade de acessos ao banco de dados.

### 4.2.1 Dependência com o RUP

A Versão Alfa foi toda baseada no *Rational Unified Process*. Todas as metodologias criadas no Methodology Explorer, obrigatoriamente seriam compostas por fluxos, atividades, passos, papéis e artefatos. Não havia possibilidade de criação (no sentido mais geral da palavra) de uma metodologia totalmente nova. Todas elas eram criadas seguindo basicamente uma mesma estruturação.



**Figura 3.** O modelo de dados lógico da Versão Alfa

Essa limitação influenciou também o projeto e a implementação do código. Conceitos como fluxos e atividades, entre outros conceitos relacionados ao RUP estavam

explicitamente representados nos seus módulos de implementação e no modelo da base de dados. A Figura 3 apresenta o modelo lógico da base de dados em que se pode ver claramente que apenas o RUP ou variações dele estava sendo representado.

Foi identificado que, para que o Methodology Explorer pudesse se tornar uma ferramenta genérica para definição de metodologias, seria preciso um esforço grande para remodelar todo o banco de dados. Além disso, como o código refletia a estrutura do banco de dados, os mesmo problema era verificado nos módulos que representavam e manipulavam os elementos persistentes.

#### **4.2.2 Acessos ao Banco de Dados**

Também foi verificado que a ferramenta utiliza muito pouco o armazenamento temporário de informações em memória. Quando o usuário realizava qualquer operação sobre uma metodologia, seja ela criando, alterando ou removendo componentes, os efeitos eram instantaneamente refletidos no banco de dados. Por conseguinte, o acesso ao banco de dados era muito freqüente.

Em geral, o acesso ao banco de dados é um dos fatores mais críticos para performance de qualquer sistema. O fato de a ferramenta não guardar dados por muito tempo em memória contribuía bastante para aumentar a freqüência de acessos ao banco, o que conseqüentemente afetava a performance do sistema como um todo.

A análise de usabilidade demonstrou que os usuários se queixaram muito da lentidão no processamento de ações. Isso foi causado em grande parte por essa questão do acesso ao banco, verificado na análise do projeto e da implementação. Assim, uma melhor implementação desse aspecto melhoraria tanto a performance do sistema como sua usabilidade.

## 5 Melhorias e Evoluções de Projeto

A realização do experimento com os usuários e a análise de seu protocolo verbal juntamente com o estudo e avaliação do projeto e implementação resultaram, como descrito no capítulo anterior, na identificação imediata de uma grande quantidade de pontos que mereceriam melhorias na nova versão a ser lançada. A análise foi de fundamental importância e serviu como ponto de partida para a reestruturação e expansão da ferramenta, o principal objetivo deste trabalho.

### 5.1 A Remodelagem Arquitetural

A arquitetura da Versão Alfa, como já descrito anteriormente, era toda baseada em elementos que correspondiam ao RUP. Isso limitava por demais o escopo da ferramenta, pois nada além de instâncias do RUP poderiam ser trabalhadas.

A arquitetura da ferramenta na versão 1.0 foi totalmente remodelada para se tornar independente de metodologia, ou seja, se tornar genérica. O Methodology Explorer agora passou a ser capaz de manipular tanto o RUP como também qualquer tipo de metodologia de desenvolvimento de software. Essa foi uma das maiores evoluções deste trabalho, pois representou uma mudança enorme na utilidade e no poder de alcance da ferramenta.

Na Versão Alfa, os tipos de componentes básicos eram metodologia, fluxo, atividade, artefato, passo, papel e guia. Qualquer componente criado tinha que obrigatoriamente ser de algum desses tipos. Imagine se o RUP evoluísse ao ponto de surgirem mais alguns conceitos novos. Como a ferramenta foi toda construída em cima dos tipos atuais, ela cairia em desuso facilmente por não poder representar os novos conceitos.

O fato de a nova arquitetura ser genérica permite a representação de qualquer conceito novo. Esse um grande diferencial: o usuário é quem define os conceitos agora. Isso garante à ferramenta uma maior longevidade na sua utilização, pois o surgimento de algum conceito novo, não só no RUP, mas em qualquer metodologia, poderia ser facilmente capturado.

### 5.1.1 Elementos Básicos do Methodology Explorer 1.0

Para representar os conceitos presentes em metodologias, o novo Methodology Explorer só impunha a restrição de que a metodologia fosse capaz de ser representada como uma hierarquia de componentes. Visto isso, para representar as metodologias de forma genérica, só seria necessário que a ferramenta trabalhasse com dois conceitos básicos: componentes e tipos de componentes. Com esses dois elementos, as árvores de componentes, que representavam as metodologias, poderiam ser criadas a contento. Para melhor representar os conceitos um tipo especial de componente, *metodologia*, também foi criado para representar as metodologias. Obviamente um dos elementos mais importantes. Como uma forma de adicionar ainda mais poder à ferramenta, dois novos conceitos também foram criados: *conteúdos anexos* e *regras de coleções*. Cada um dos quatro elementos básicos será melhor descrito a seguir.

#### 5.1.1.1 Componente

Um *componente* é o elemento que representa qualquer parte presente numa metodologia. Fazendo uma analogia com o RUP, um fluxo seria um *componente*, um artefato seria um *componente*, assim como um papel também seria um *componente*.

Para representar aqueles elementos do RUP, o usuário não cria mais um fluxo ou uma atividade, agora ele cria um *componente* de forma genérica. A semântica que cada *componente* representa, ou seja, aquilo que diz se ele é um fluxo, uma atividade ou qualquer outra coisa, deve ser dada pelo tipo do componente, o próximo elemento a ser discutido.

Um *componente* traz consigo uma série de informações como seu título, descrição e seu *conteúdo anexo* (este será discutido mais a frente). Além dessas informações, um *componente* também agrega conjuntos de outros componentes na sua estrutura. Por exemplo, no RUP, uma atividade possui passos. Assim, como uma atividade é um *componente* e um passo também é um *componente*, significa que um *componente* deve possuir conjuntos de outros *componentes*.

#### 5.1.1.2 Tipo de Componente

Um *tipo de componente* é um elemento que serve para identificar características comuns a vários *componentes*. Cada *componente* deve ter um tipo que o represente. Por exemplo, os *componentes* Requisitos, Implementação e Testes são *componentes* cujo tipo de *componente* é Fluxo. Os *componentes* Avaliar Iteração e Desenvolver Plano de Negócio são *componentes* do tipo Planejamento e Gerenciamento.

O usuário pode definir diversos *tipos de componentes*. Como já destacado anteriormente, é esse elemento que permite que a ferramenta possa facilmente ser usada para representar qualquer conceito. Para isso, basta definir um novo tipo.

Um tipo é particularmente útil para que um componente possa discriminar os diferentes conjuntos de componentes que ele contenha. Vimos que um componente possui outros componentes. Com a definição desse elemento, podemos dizer agora que um componente deve possuir conjuntos de componentes de diferentes tipos.

Traduzindo isso para um exemplo do RUP, podemos dizer que Requisitos é um *componente* do tipo Fluxo e possui, como sub-componentes, *componentes* do tipo Atividades. Um desses *componentes*, Encontrar Atores e Use Cases, é um *componente* do tipo Atividade e que possui, dentro de si, um conjunto de *componentes* ordenados do tipo Passo, um conjunto de *componentes* do tipo Artefatos de Entrada e um conjunto de *componentes* do tipo Artefatos de Saída.

Um tipo deve conter informações como seu título e descrição, mas também deve ser capaz de armazenar algumas regras pertinentes à natureza dos *componentes*. No exemplo acima, observe que é dito que a atividade não possui simplesmente passos, mas sim, passos ordenados. O fato de o *tipo de componente* ser ordenado é representado por *regras de coleção* definidas no *tipo de componente*. *Regras de coleção* serão abordadas mais adiante.

#### 5.1.1.3 Metodologia

O elemento *metodologia* representa o conceito central do Methodology Explorer: as metodologias. A *metodologia* nada mais é do que um tipo particular de *componente* usado como elemento raiz da árvore hierárquica de componentes que compõem cada metodologia.

#### 5.1.1.4 Conteúdo Anexo e Regras de Coleção

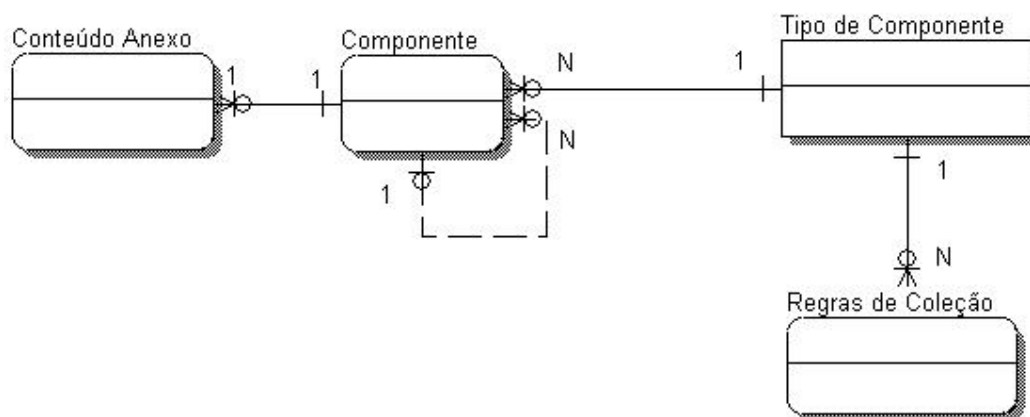
*Conteúdo anexo* e *regras de coleção* são elementos que fazem parte dos elementos *componente* e *tipo de componente*, respectivamente. Um *conteúdo anexo* representa, como o próprio nome indica, o conteúdo de um *componente*. Este conteúdo tanto pode ser um texto ou como um arquivo. A possibilidade de anexar arquivos aos componentes é uma nova funcionalidade incluída na nova versão do Methodology Explorer.

As *regras de coleção* são utilizadas para atribuir semântica às coleções de componentes. Como cada coleção é representada por um elemento *tipo de componente*, as *regras de coleção* são definidas para cada *tipo de componente*.



### 5.1.2 Modelagem dos Dados

A definição de uma arquitetura genérica teve também, além da criação dos elementos citados anteriormente, uma remodelagem total da base de dados. A Figura 4 mostra como foi redefinido o modelo de dados.



**Figura 4.** O novo modelo de dados da versão Methodology Explorer 1.0

Comparando esse o modelo com o apresentado na Figura 3 (página 13), percebe-se claramente que este é um modelo bem mais simples. Apesar de mais simples, ele é capaz de representar conceitos genéricos, o que não era possível antes.

## 5.2 Acesso Controlado à Base de Dados

Um dos fatores que mais desagradavam os usuários era a lentidão do processamento. A análise da implementação demonstrou que grande parte do tempo era gasto com o acesso ao banco de dados, feito praticamente a cada ação do usuário.

Para resolver esse problema, o acesso ao banco de dados agora passou a ser racionalizado. As alterações que o usuário realiza na metodologia de trabalho agora são mantidas em memória até que o usuário determine sua confirmação. No ato da confirmação, os dados são persistidos no banco de dados.

O uso da memória, da forma como descrita acima, diminui bastante a quantidade de acessos ao banco de dados e, conseqüentemente, as operações passam ser realizadas mais rapidamente. Com isso, melhora a usabilidade do sistema e a própria produtividade dos usuários.

A solução de guardar alterações temporariamente em memória fez surgir um novo conceito na ferramenta: o conceito de “salvamento” da metodologia. Quando o usuário está trabalhando com uma metodologia do Methodology Explorer, a metodologia é considerada não salva até que o usuário salve as alterações, o que pode ser feito através

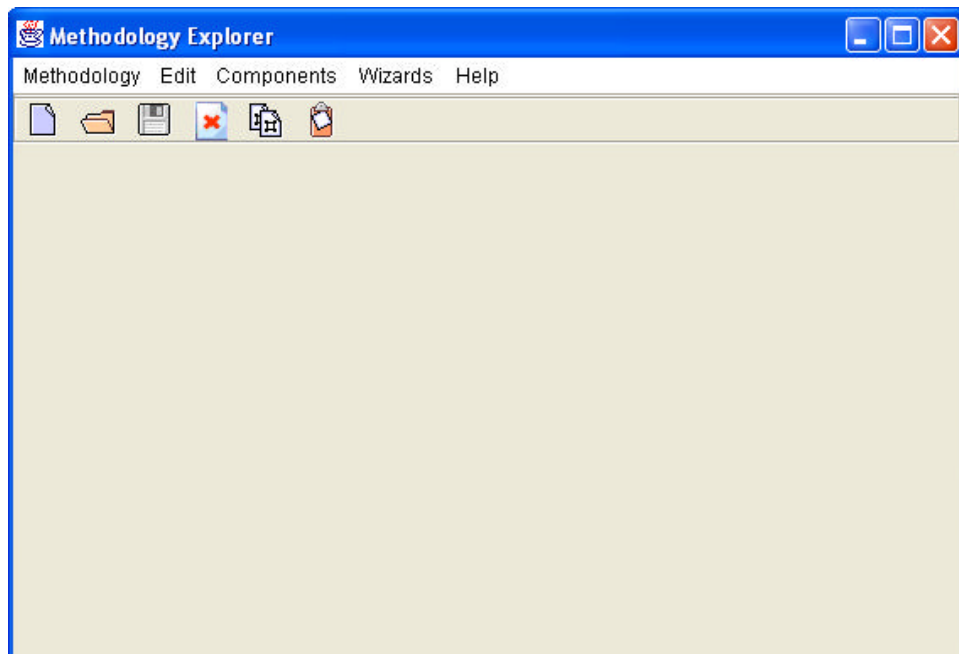
do menu ou de um atalho na barra de ferramentas. O conceito de metodologia salva assumiu um papel importante, tanto que algumas funcionalidades novas, como publicação em HTML e exportação da metodologia, têm como pré-condição que a metodologia esteja salva.

## 6 O Novo Design da Interface Gráfica

A interface gráfica é uma das partes mais importantes de qualquer sistema. Através dela os usuários podem interagir com sistema e obter deste os resultados esperados. Mais do isso, se olharmos apenas do ponto de vista do usuário, a interface gráfica representa “o sistema” por ser exatamente o que o usuário enxerga.

A usabilidade do Methodology Explorer, analisada para sua Versão Alfa, demonstrou ser muito baixa. A análise também mostrou que grande parte dos problemas identificados se referiam ao mau projeto de sua interface. Um bom projeto de interface é fundamental para que uma ferramenta possa ser aceita pelo menos como útil por quem a deve utilizar. Por esse motivo, para a versão 1.0, foi necessário realizar definir uma interface capaz não só de permitir a realização das tarefas dos usuários, mas também capaz de atender suas exigências.

A reestruturação dessa interface teve como objetivo corrigir os problemas de usabilidade já identificados e deixá-la sofisticada o suficiente para poder oferecer eficientemente as novas funcionalidades e recursos pretendidos na versão 1.0. A Figura 5 traz a nova tela inicial.



**Figura 5.** A tela inicial do Methodology Explorer 1.0

A seguir será abordado o que foi realizado para alcançar o primeiro objetivo, ou seja, a melhoria da usabilidade. Os detalhes da interface decorrentes das novas funcionalidades serão vistos quando estas forem abordadas num capítulo posterior.

## 6.1 Atalhos e a Diversificação de Opções para Realização de Tarefas

Anteriormente, algumas tarefas na ferramenta necessitavam de vários cliques de mouse na árvore de componentes e nos itens de menu para poderem ser realizadas. Além de o processamento da interface já ser lento, a quantidade grande de cliques contribuía para aumentar o tempo necessário para a realização das atividades.

Foi observado também que a ferramenta só dispunha de uma única seqüência de passos para completar cada atividade. Isso significa que não existia qualquer forma alternativa para o usuário escolher como realizar seu trabalho. Cada recurso era acessado da mesma forma. Na nova versão do Methodology Explorer os atalhos se tornaram um recurso bem presente e importante.

### 6.1.1 Barra de Ferramentas

Um dos atalhos principais é a barra de ferramenta. Esta possui botões que representam atalhos para as principais funcionalidades do sistema. Cada funcionalidade é identificada por ícones que lembram as funcionalidades em questão. Através da barra, o usuário pode realizar disparar algumas operações sem a necessidade de abrir os menus existentes. Este pode também, se quiser, continuar trabalhando apenas com os menus, já que a barra de ferramenta só disponibiliza um caminho mais curto para tarefas que possuem outras formas de acesso.



**Figura 6.** Barra de ferramentas do Methodology Explorer

Na Figura 6, as funcionalidades presentes são, da esquerda para direita: Nova Metodologia, Abrir Metodologia, Salvar Metodologia, Fechar metodologia, Copiar e Colar.

### 6.1.2 Menus de *Pop-up*

Os menus de *pop-up* também são novidades desta versão. Esses menus também disponibilizam atalhos para algumas funcionalidades acessíveis através dos menus

tradicionais. A inclusão dos menus de *pop-up* fornece mais opções para o usuário e conseqüentemente melhora a usabilidade como um todo.



**Figura 7.** Uso de um menu de *pop-up*

Os menus de *pop-up* são úteis para realização de operações sobre componentes das árvores de metodologias. Normalmente, são operações que necessitam selecionar um componente e escolher uma opção de menu. Com os *pop-ups*, basta clicar com o botão direito sobre um componente e escolher a opção desejada no *pop-up* que se abrirá dependendo do tipo de componente selecionado, diminuindo sensivelmente o tempo necessário para realizar o comando.

### 6.1.3 Teclas de Combinação

Alguns comandos agora podem ser realizados através da combinação de teclas. Alguns exemplos são as operações de copiar e colar. Para copiar um componente, basta selecioná-lo e usar a combinação *control*+ *C*. Analogamente, para colá-lo basta selecionar o elemento destino e teclar *control*+ *V*.

As operações de “copiar” e “colar”, aliás, são as que possuem a maior quantidade de formas diferentes de serem realizadas: quatro. Ambas podem ser realizadas pelo menu principal, pela barra de tarefas, pelo menu de *pop-up* ou por combinação de teclas.

## 6.2 Separação Lógica de Conceitos

Existia um grave problema de invisibilidade conceitual no Methodology Explorer. Na mesma tela eram apresentadas metodologias já criadas e metodologias que estavam sendo criadas pelo usuário naquele momento. O usuário também podia manipular ao mesmo tempo todas as metodologias visíveis na tela. Além disso, todas as árvores de metodologia possuíam seus componentes sem nada que destacasse visualmente um componente de outro ou uma metodologia de outra.

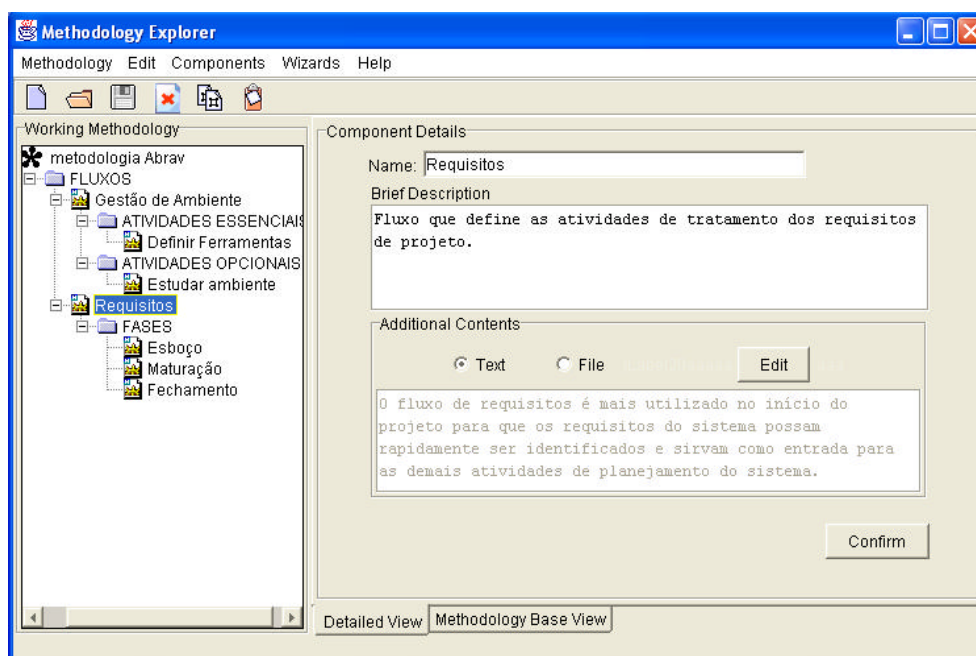
### 6.2.1 Metodologia de Trabalho

Na versão 1.0, foi criada noção de **metodologia de trabalho**. Esta seria a metodologia que o usuário estaria trabalhando num determinado momento. Os detalhes de cada componente passaram agora a só poderem ser vistos para a metodologia de trabalho em questão. Esta passou a ter um maior destaque visual, sendo inclusive apresentada num painel exclusivo, separado das demais metodologias.

### 6.2.2 Divisão da Tela em Painéis Conceituais.

Uma das maiores necessidades da ferramenta era a quantidade de conceitos apresentados numa mesma tela. A reestruturação da interface atacou esse problema dividindo melhor a tela em áreas conceituais.

Já foi dito acima que o conceito de metodologia de trabalho foi representado num painel destacado à esquerda da tela. Os demais conceitos passaram a ocupar o espaço restante da tela, no caso, meio e lateral direita. Este espaço foi melhor aproveitado através da utilização de abas. Cada aba contém um painel que é aberto ocupando todo o espaço restante da tela. O simples clique no nome das abas permite alternar a visualização das mesmas.



**Figura 8.** Tela dividida conceitualmente: metodologia de trabalho e visão detalhada

A Figura 8 mostra o Methodology Explorer e a divisão da tela em painéis. Do lado esquerdo encontra-se a metodologia de trabalho. No lado esquerdo, a aba selecionada é a que apresenta a visão detalhada de elementos da metodologia de trabalho. Na visão

detalhada, são apresentadas informações sobre qualquer elemento selecionado na árvore de componentes da metodologia de trabalho. O usuário pode visualizar e também pode livremente alterar os valores dessas informações.

A outra aba presente é a aba das metodologias (Figura 9), que mostra as metodologias já criadas na ferramenta. Os componentes dessas várias metodologias podem ser usados para compor a metodologia de trabalho que está sendo construída. As operações de “copiar” e “colar” são úteis para tal propósito: copia-se um componente de alguma dessas metodologias e cola-o na árvore da metodologia de trabalho.

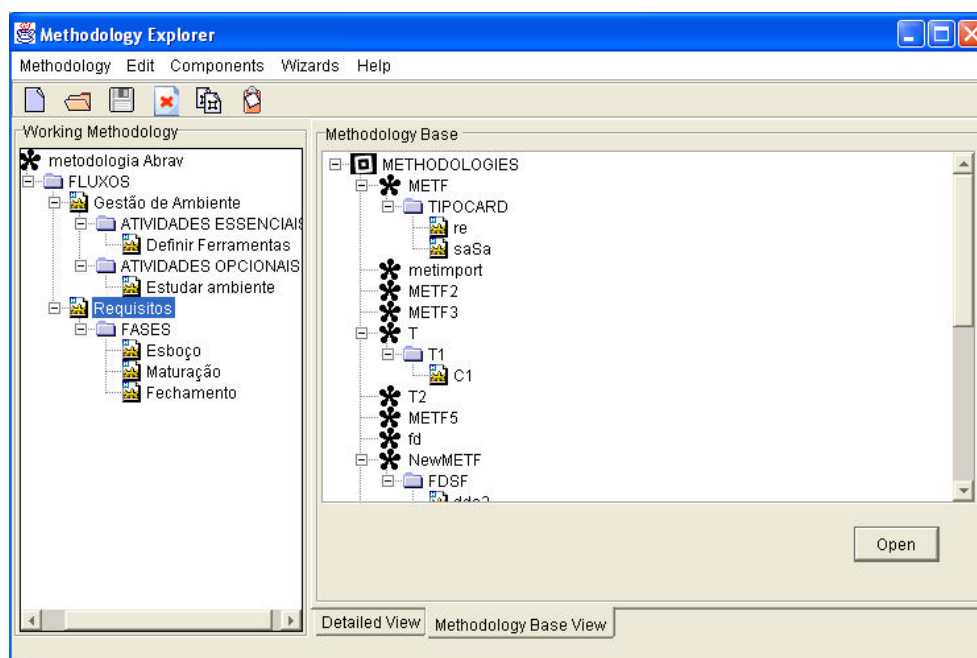
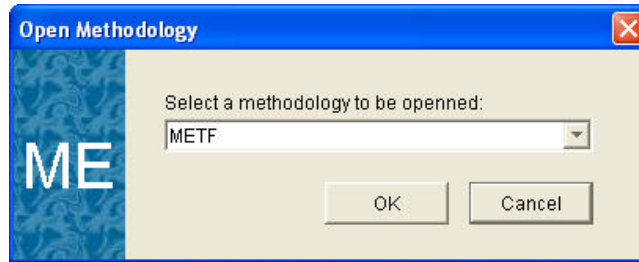


Figura 9. A visão das metodologias

### 6.2.3 Conceito de Abertura e Fechamento de Metodologias

Com a criação dos painéis, conceitos novos como “abrir” e “fechar” metodologias também surgiram. A operação de “abrir” é usada para abrir uma metodologia e torná-la a metodologia de trabalho. Analogamente, “fechar” deve ser utilizada para remover a apresentação da metodologia de trabalho que está aberta. Uma vez aberta, a metodologia é apresentada no painel destacado a esquerda da tela.

Para abrir uma metodologia deve-se acessar o menu *Methodology* e acessar o item *Open*, ou utilizar a barra de tarefas. Essa seqüência abrirá uma tela para que o usuário selecione a metodologia que será aberta como apresentado abaixo.



**Figura 10.** Abertura de uma metodologia.

Para fechar a metodologia, podem ser usadas as mesmas opções utilizadas para abertura: menu ou barra de ferramentas. O fechamento fecha todos os painéis e deixa a tela como mostrado na Figura 5 (página 20).

### **6.3 Mapeamento de Funcionalidades em Controles Gráficos**

A nova interface gráfica também foi contemplada com uma melhor distribuição dos controles gráficos responsáveis por acionar as funcionalidades. Um controle é algum elemento gráfico, como um botão, um menu ou caixa de texto cuja função é claramente visível ao usuário. Além disso, segundo [8], a usabilidade melhorar quando cada funcionalidade possui um controle que é usado exclusivamente para a realização daquela funcionalidade. A justificativa para isso é que o usuário pode facilmente usar o controle sem ter dúvida da funcionalidade a que ele se destina. É importante também que os controles sejam facilmente identificáveis, ou seja, que possuam alguns recursos, como um texto ou um ícone, que facilitem seus reconhecimentos e seus objetivos.

As modificações realizadas na interface gráfica seguiram as diretrizes propostas acima. O estudo do protótipo demonstrou ausência de ícones nos elementos que eram dispostos na tela. Além disso, foi verificado que algumas funcionalidades até não possuíam nenhum controle gráfico explícito. O objetivo passou a ser corrigir ao máximo os problemas identificados e seguir diretrizes que levassem a uma evolução planejada da interface e, conseqüentemente, também de todo o Methodology Explorer.

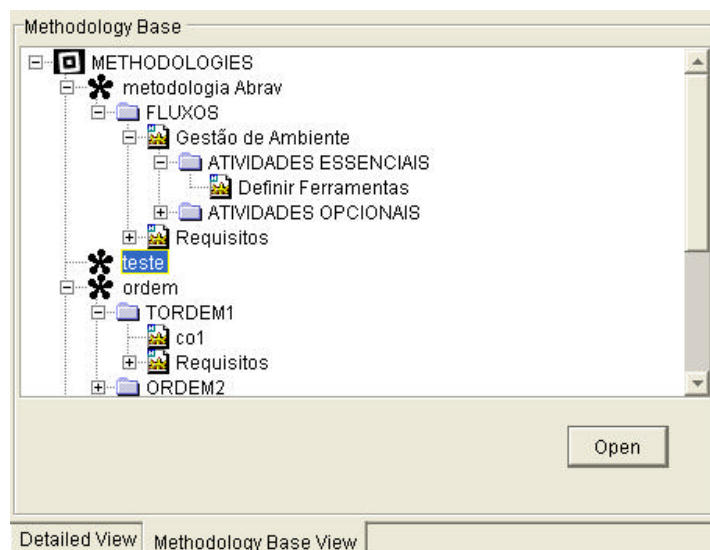
#### **6.3.1 Uso de Ícones**

Os ícones são elementos muito úteis para identificar funcionalidades e elementos. Quando bem selecionados, eles facilitam bastante a memorização das operações associadas a cada controle. A partir dessa nova versão, a interface gráfica passa a contar fortemente com esse recurso.

Uma das aplicações mais importantes dos ícones é na distinção entre os elementos presentes nas árvores de metodologias. Uma árvore possui os elementos metodologia,



componente e tipo de componente. Como cada nó da árvore apresentava apenas o título do elemento, ficava difícil identificar qual o tipo de cada elemento. Na nova versão, cada elemento da árvore, além do seu próprio título, agora passa a ter também um ícone diferente associado, facilitando o entendimento e a compreensão dos usuários.



**Figura 11.** As árvores de metodologias com nós iconizados

A Figura 11 apresenta as árvores de metodologias como seus nós expandidos, onde se podem ver os ícones presentes em cada nó. Um elemento *metodologia* é representado pelo ícone . Cada elemento *metodologia* é raiz de sua árvore (cada árvore de metodologia é iniciada por um elemento *metodologia*). Abaixo de *metodologia* aparecem *tipos de componente*, representados pelos ícones , e abaixo dos tipos aparecem os *componentes* que possuem os ícones . Percebe-se também que os *componentes* agrupam *tipos de componentes* como nós filhos e que os tipos possuem *componentes* como nós filhos e assim por diante.

No painel da visão das metodologias, percebe-se ainda um nó que é usado para agrupar as metodologias através de uma única raiz. Esse elemento não possui nenhum valor semântico nas metodologias, a não ser o de agrupá-las graficamente com uma única raiz. O ícone para essa raiz é .

Além de facilitar o reconhecimento dos elementos, os ícones também são usados como recurso para melhorar a aparência da ferramenta, como os ícones usados na barra ferramentas (Figura 6, página 21).

### 6.3.2 Criação de Novos Controles

Algumas funcionalidades do Methodology Explorer não possuíam controles gráficos explícitos. Por exemplo, nos concentremos no conceito de instanciação de metodologia.

Uma instanciação consiste em criar uma nova metodologia a partir de uma outra metodologia já existente. A nova metodologia criada deverá possuir todos os seus elementos também presentes na metodologia que originou a instanciação. Na Versão Alfa do Methodology Explorer não existia explicitamente a funcionalidade “instanciar metodologia”, apesar de que, se o usuário resolver copiar os componentes de uma metodologia para uma nova que ele está criando, ele estaria realizando uma instanciação. O que se pretende mostrar é que havia conceitos importantes que estavam “mascarados” dentro da ferramenta que necessitavam de um tratamento mais explícito.

Além da instanciação, outros conceitos também não possuíam controles gráficos que mapeassem nas funcionalidades pretendidas. Esses problemas foram resolvidos através da criação de opções claras na ferramenta para suas utilizações, como botões ou itens de menus para acionar cada funcionalidade. A funcionalidade de instanciação, por sua vez, pôde ser representada através de *wizard*, ou seja, uma janela que mostrava o passo-a-passo necessário para realizar a instanciação de uma metodologia. Os *wizards* criados serão abordados no próximo capítulo.

## 7 Os Novos Recursos e Funcionalidades

Nos capítulos anteriores detalhamos uma série de evoluções realizadas no Methodology Explorer. Elas tiveram como objetivo, principalmente, corrigir algumas deficiências e melhorar alguns aspectos de usabilidade e navegabilidade na interface gráfica. De agora em diante iremos abordar uma série de aspectos novos [9] que foram incorporados à ferramenta. Esses aspectos se constituem em um conjunto de recursos considerados essenciais para uma ferramenta que se propõe a realizar tarefas do Methodology Explorer.

Não basta apenas criar as metodologias e reusar componentes. É preciso oferecer algo mais ao projetista de metodologias ou ao engenheiro de processos. Uma metodologia não é composta apenas de texto, mas também de guias, *templates* e outros arquivos. A partir de agora, a funcionalidade de anexação de arquivos também está presente.

Além da anexação de arquivos, foi criado o mecanismo de importação e exportação de metodologias. Este é capaz exportar uma metodologia criada na ferramenta para o sistema de arquivos e representar todos os seus elementos e ligações entre eles. Da mesma forma, essa representação pode ser importada para a ferramenta e exibida como se tivesse sido criada como outra metodologia qualquer. Muito poderoso!

Publicação também é algo essencial. Ninguém cria uma metodologia sem ter como objetivo apresentá-la. O Methodology Explorer agora permite publicar uma metodologia criada na ferramenta no formato HTML. Mais precisamente permite gerar um *site* para a metodologia. Isso é feito a partir de uma pequena ferramenta integrada, um *wizard* chamado *Methodology Explorer Web Publisher*.

O *Web Publisher* não foi o único criado, para outras funcionalidades importantes também foram criadas algumas ferramentas como essa. Uma funcionalidade discutida anteriormente, a instanciação de metodologias, não existia explicitamente na ferramenta. Agora ela pode ser realizada através da ferramenta *Methodology Explorer Instance Creator*. Outra ferramenta criada foi o *Methodology Explorer Analyser*. Esse *wizard* funciona como um compilador de componentes, tendo o propósito de comparar metodologias e gerar como respostas análises de compatibilidade entre as mesmas.

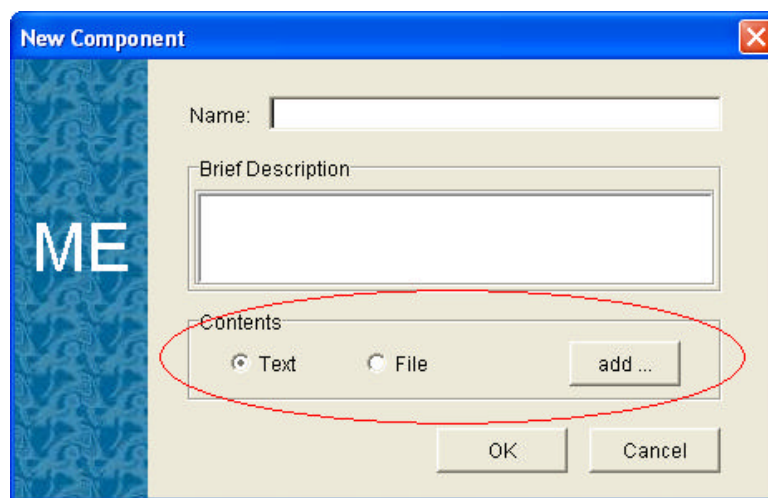
A seguir, esses novos recursos serão mais detalhados.

## 7.1 Anexação de Arquivos

O elemento *componente* é usado para representar qualquer parte de uma metodologia. Nas árvores de metodologias, esse elemento aparece após o elemento *metodologia*, sempre agrupados através do elemento *tipo de componente*, que agrupa coleções de componentes de um mesmo tipo.

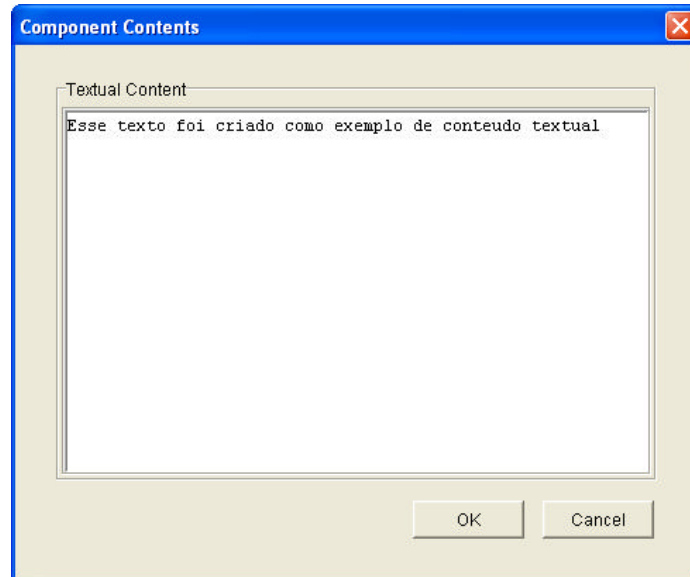
Um *componente* armazena uma série de informações relacionadas a si. As informações se constituem de meta-informações, como título e descrição, e do conteúdo propriamente dito. Agora são oferecidas duas opções de conteúdo de componentes: conteúdo textual ou conteúdo arquivo.

No momento da criação de um componente, o usuário deve informar, obrigatoriamente, o tipo de conteúdo será inserido, como mostrado na Figura 12. Não é possível criar um componente sem conteúdo. A tela de criação do componente é aberta quando o usuário seleciona um tipo de componente (indicando o local onde o componente será inserido) e segue clicando nos menus *Components -> Add Component* ou utiliza um menu de *pop-up* (Figura 7).



**Figura 12.** Escolha do tipo de conteúdo na criação de um componente

A opção de conteúdo textual é aquela em que o usuário apenas necessita digitar um texto para preencher o componente. Esse texto é digitado na própria ferramenta. Caso a opção de conteúdo textual seja escolhida, a tela de inserção de conteúdo textual será apresentada Figura 13, quando usuário clica no botão que possui o texto “add ...” da Figura 12.



**Figura 13.** Inserção de conteúdo textual

A opção para conteúdo arquivo é um recurso criado nesta versão que adiciona um poder extra à ferramenta. Ao escolher esta opção, uma caixa de diálogo será aberta para que o usuário selecione, no seu sistema de arquivos, aquele arquivo que será anexado como o conteúdo componente.

Os *bytes* do arquivo são inteiramente armazenados dentro do Methodology Explorer, ou seja, não é armazenado apenas um *link*.

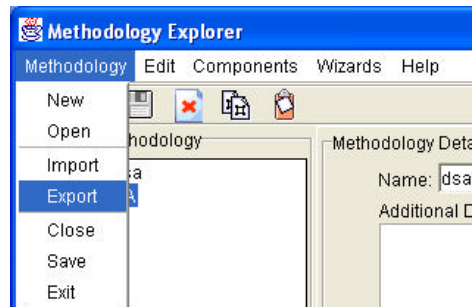
## 7.2 Importação e Exportação de Metodologias

A funcionalidade de importação e exportação de metodologias permite que uma metodologia inteira seja transportada para o sistema de arquivos. Os mesmos dados do sistema de arquivos, posteriormente podem ser importados para o Methodology Explorer novamente. Essa funcionalidade tem grande importância nesta nova versão, pois permite troca de informações entre os membros um grupo de trabalho. Uma metodologia criada por um usuário, numa máquina A, pode ser exportada para o sistema de arquivos e levada para uma máquina B, onde poderá ser importada por inteiro. Esse recurso não existia anteriormente.

Tanto na importação quanto na exportação, são trafegados todos os dados da metodologia escolhida, inclusive seus conteúdos anexados como arquivo. A exportação de uma metodologia gera um arquivo no formato XML capaz de representar toda a hierarquia de componentes presentes na metodologia. XML [11] vem de *eXtended Markup Language* e é uma tecnologia que permite definir de forma textual hierarquias elementos seguindo um padrão definido pelo próprio usuário da tecnologia. Esta

tecnologia tem crescido bastante nos últimos anos e tem se tornado um padrão mundial para troca de informações entre sistemas heterogêneos ou não.

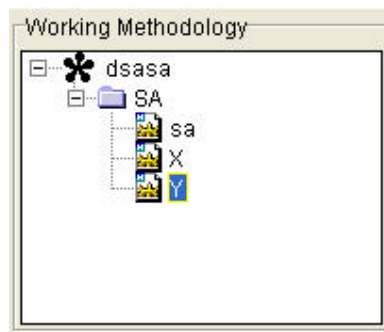
O arquivo XML gerado, além da hierarquia de componentes, consegue representar também as informações presentes nos elementos: as meta-informações e os conteúdos textuais. Se a metodologia exportada possuir componentes com conteúdos arquivos, são geradas cópias desses arquivos no sistema de arquivos. Os arquivos criados no disco mantêm-se com os mesmos nomes de quando foram anexados ao componente.



**Figura 14.** Item de menu para exportação de metodologia

Através do menu *Methodology* -> *Export* (Figura 14), dá-se início o processo de exportação. Quando clicado este menu, o usuário deverá indicar o diretório onde serão gerados os arquivos de exportação e, em seguida, a ferramenta irá proceder à exportação da metodologia de trabalho corrente. No diretório selecionado, será gerado um subdiretório como o mesmo nome da metodologia e, dentro deste, o arquivo XML que representa a metodologia. Nesse mesmo diretório, serão gerados diretórios de mesmos nomes dos componentes que possuem conteúdos com arquivos anexados. Nestas pastas estarão os respectivos arquivos.

O exemplo abaixo mostra um exemplo de exportação. A metodologia “dsasa” possui os componentes “X” e “Y” com conteúdos arquivos.



**Figura 15.** Metodologia sendo exportada

A exportação da mesma, no diretório “c:\meu\_diretorio”, gerará a seguinte estrutura de arquivos e diretórios, no sistema de arquivos:



**Figura 16.** A metodologia exportada no sistema de arquivos

Para fazer a importação da metodologia, será necessário que a mesma estrutura de diretórios seja mantida, ou seja, não deve haver modificação dentro dos diretórios criados. Na verdade, pode até haver alteração desde que essa alteração seja refletida no arquivo XML. Por exemplo, este pode referenciar um arquivo X que representa um conteúdo de um componente Y. Caso o nome do arquivo M seja modificado, no sistema de arquivos, para N, então o usuário deverá alterar o arquivo XML para que ele agora referencie o arquivo chamado N. O mesmo vale para as demais alterações: caso ocorram, devem ser refletidas no arquivo XML da metodologia.

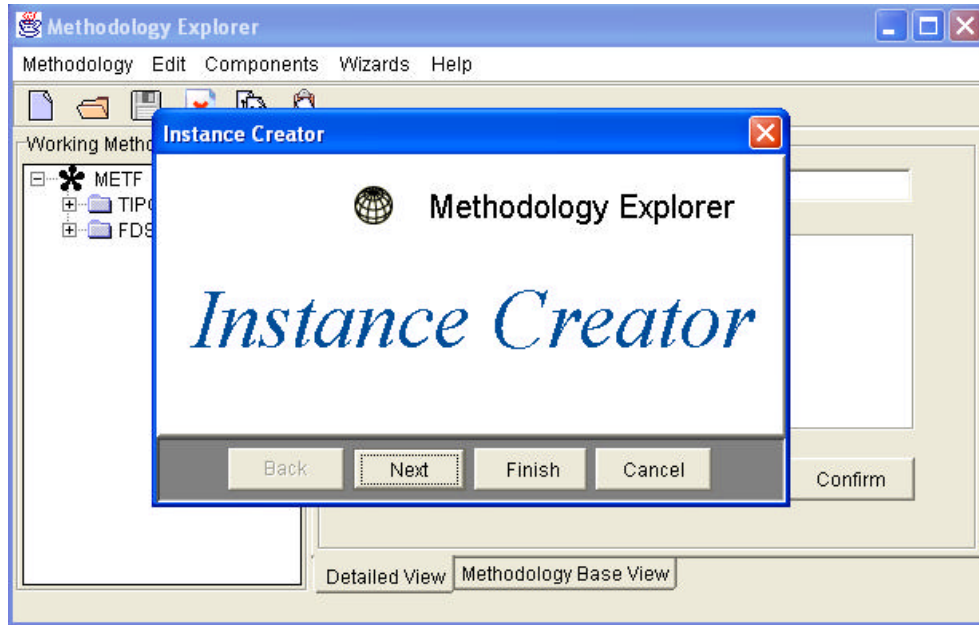
Para importar uma metodologia, o usuário deve seguir passos análogos ao da exportação. Primeiro deverá iniciar pelo menu *Methodology* e escolher a opção *Import*. Em seguida uma caixa de diálogo será aberta para que seja escolhido o arquivo que contém a metodologia. Neste caso, basta o arquivo XML da metodologia ser escolhido, o que resultará na importação dos componentes e dos arquivos anexados. Ao fim, a metodologia importada será apresentada como metodologia de trabalho atual.

Os próximos recursos apresentados neste capítulo mostrarão algumas funcionalidades que foram implementadas através da criação de sub-ferramentas do *Methodology Explorer*, todas elas acessíveis a partir do menu *Wizards* da barra de menu.

### 7.3 Methodology Explorer Instance Creator

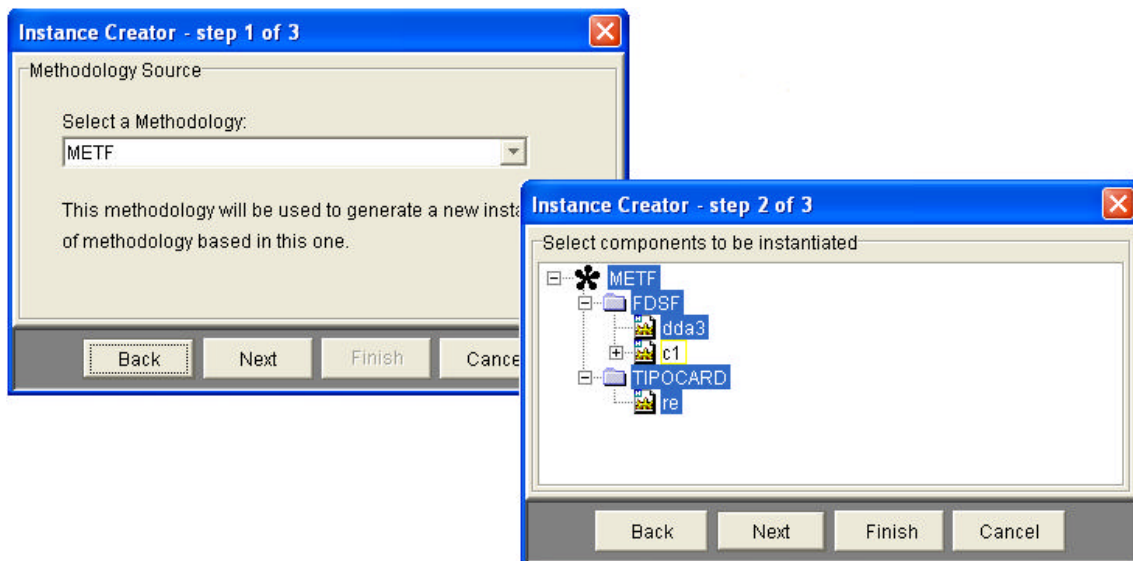
O *Methodology Explorer Instance Creator* é um *wizard* criado para guiar o usuário na realização da atividade de **instanciação** de uma metodologia. Por instanciação se entende a definição de uma nova metodologia a partir de uma metodologia base. A nova metodologia é dita instância da metodologia base e todos os seus componentes e tipos devem estar presentes na metodologia base. Em outras palavras, uma instância é uma metodologia composta por um subconjunto de componentes de uma outra.

O *wizard* de instanciação é iniciado através do menu *Wizards -> Instance Creator*.



**Figura 17.** *Methodology Explorer Instance Creator*

No passo 1, o usuário informa qual metodologia será instanciada e no passo seguinte deve selecionar os componentes que farão parte da nova metodologia sendo criada.



**Figura 18.** Alguns passos do *Instance Creator*

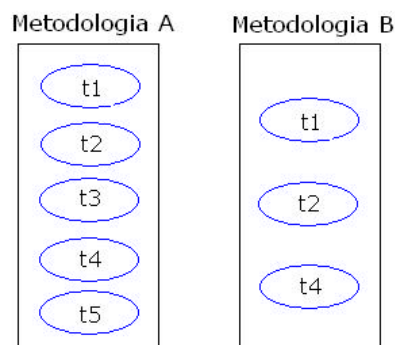
Neste ponto, já pode ser finalizado o processo. No último passo, passo 3, caso o usuário ainda não tenha finalizado, pode-se escolher um nome para metodologia diferente do nome sugerido pelo *Instance Creator*. A instanciação criará uma nova metodologia instanciada a partir da metodologia base escolhida, contendo os componentes selecionados. Esta será aberta como nova metodologia de trabalho.



## 7.4 Methodology Explorer Analyser

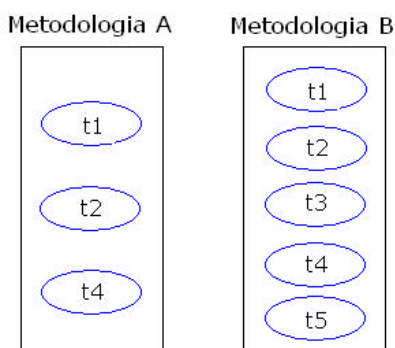
O *Methodology Explorer Analyser* é uma ferramenta que vem complementar a noção de instanciação. Ela tem o propósito de comparar metodologias através da análise de seus tipos principais. O usuário deve apenas informar duas metodologias: uma fonte e uma base. O objetivo da ferramenta é apresentar como resultado a comparação da metodologia fonte com a metodologia base, ou seja, qual a relação entre a metodologia fonte e a base.

Existem dois tipos de relações definidas para as metodologias. Já vimos uma: a instanciação, que afirma que uma metodologia é uma instância de outra quando os tipos de seus componentes formam um subconjunto dos tipos presentes na metodologia base.



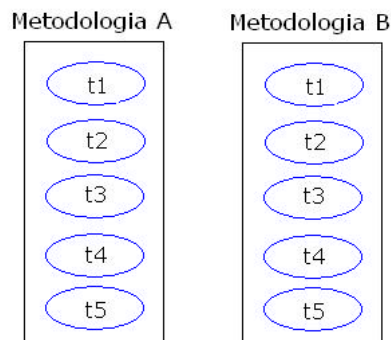
**Figura 19.** Noção de instância: B é uma instância de A

A outra relação definida é a de variante. Uma metodologia é dita variante de outra quando os seus tipos englobam todos os tipos presentes na outra metodologia. Como se pode observar, as definições de instância e variante possuem uma ligação entre si: se uma metodologia A é uma instância de B, então B é uma variante de A. Isso fica claro, pois as duas definições utilizam a noção de subconjunto.



**Figura 20.** Noção de variante: B é uma variante de A

Pode ocorrer o caso especial em que os subconjuntos são iguais, ou seja, quando as duas metodologias possuem exatamente os mesmos tipos. Neste caso, uma metodologia A seria uma instância e variante de B e vice-versa.

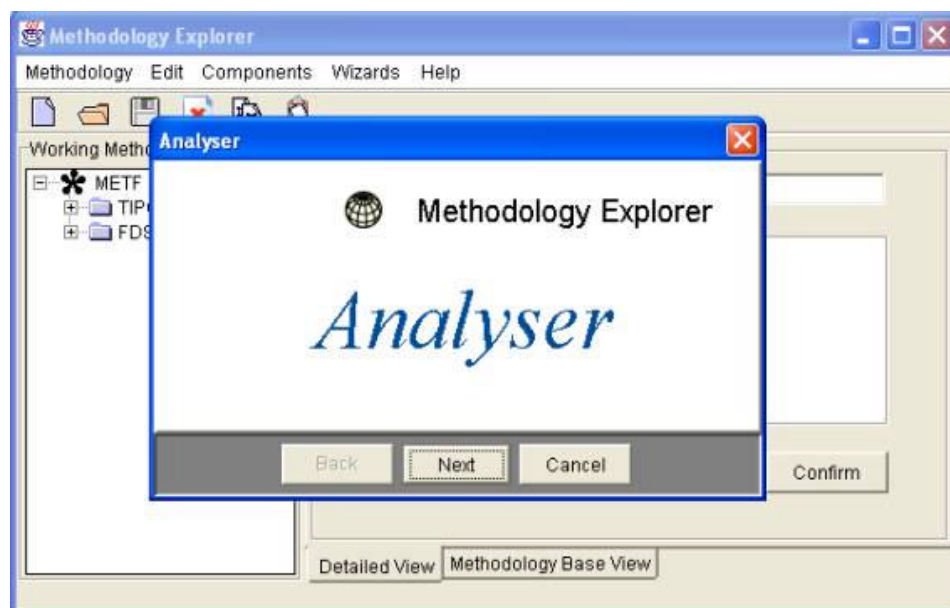


**Figura 21.** A é instância e variante de B e vice-versa

O *Analyser* deve gerar como resultado a relação existente entre as metodologias. Mais precisamente, a relação da metodologia fonte em relação à metodologia base. Assim quatro resultados são possíveis:

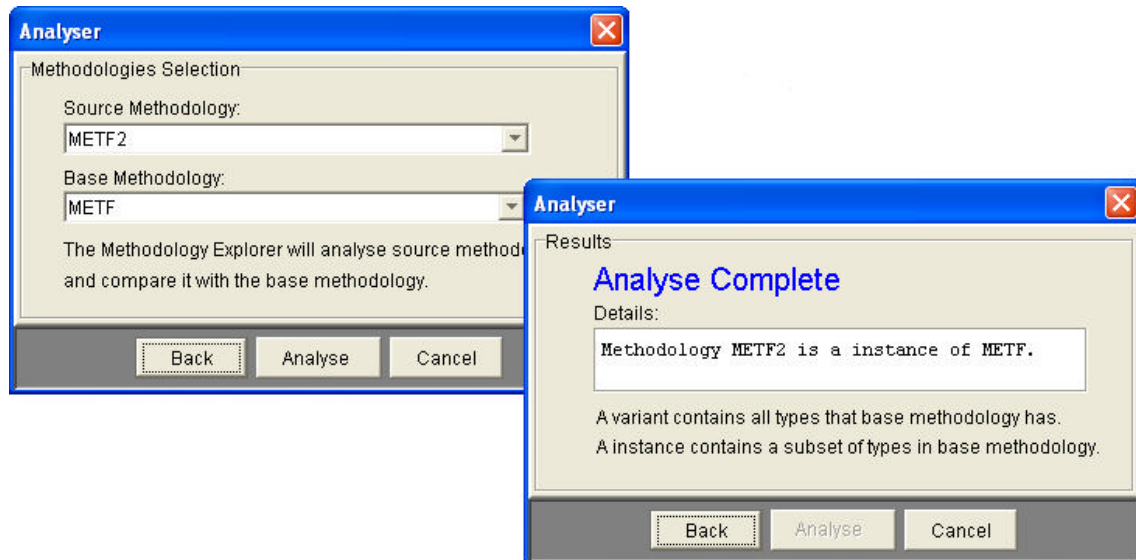
- A é instância de B;
- A é variante de B;
- A é instância e variante de B;
- A e B não estão relacionadas.

O *Analyser* é iniciado através do menu *Wizards -> Analyser*.



**Figura 22.** Methodology Explorer Analyser

O passo inicial é seleccionar as metodologias fonte e base e, logo em seguida, clicar no botão *Analyse* que imediatamente apresentará o resultado da análise.



**Figura 23.** Passos do *Analyser*

O *Analyser* se baseia nos tipos presentes nas duas metodologias. O critério utilizado para definir quando dois tipos são equivalentes leva em consideração as regras de coleção presentes nos tipos. Os nomes dos componentes e dos tipos não são considerados na análise. As regras basicamente definem algumas condições das coleções de componentes como, por exemplo, a cardinalidade (quantidade de componentes que compõem a coleção) e se a coleção é ordenada ou não. Como cada coleção possui componentes e um mesmo tipo, as regras fazem parte de cada tipo.

## 7.5 Methodology Explorer Web Publisher

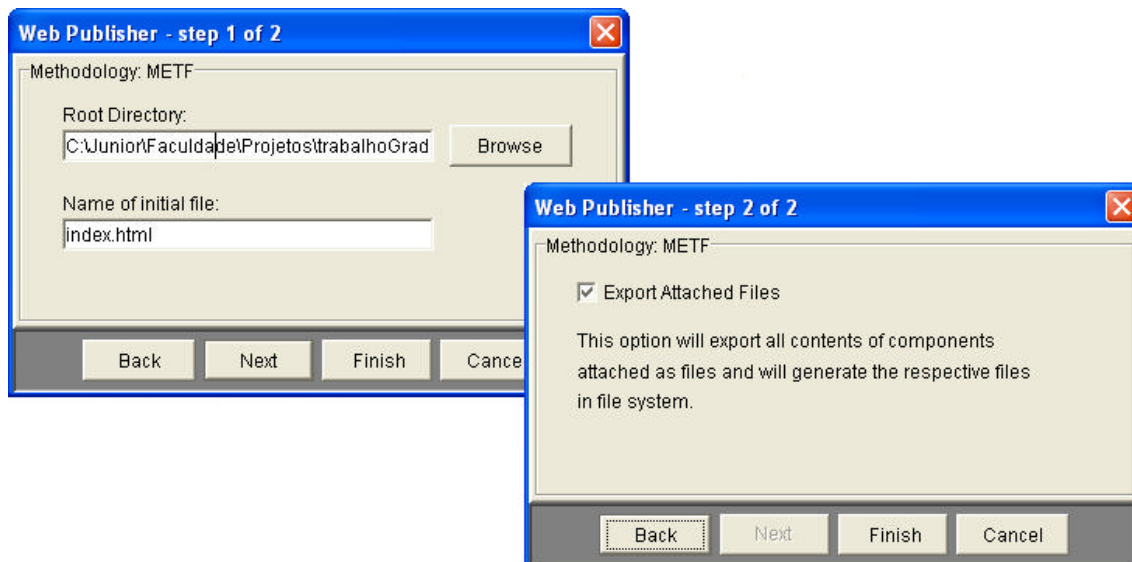
O terceiro *wizard* criado é responsável por uma das mais importantes funcionalidades acrescentadas ao Methodology Explorer: a publicação de metodologias em HTML. O *Methodology Explorer Web Publisher* permite a geração de um *site* para a metodologia de trabalho corrente possibilitando a sua publicação na *web*.

O *wizard* é iniciado através do menu *Wizards -> Web Publisher*.



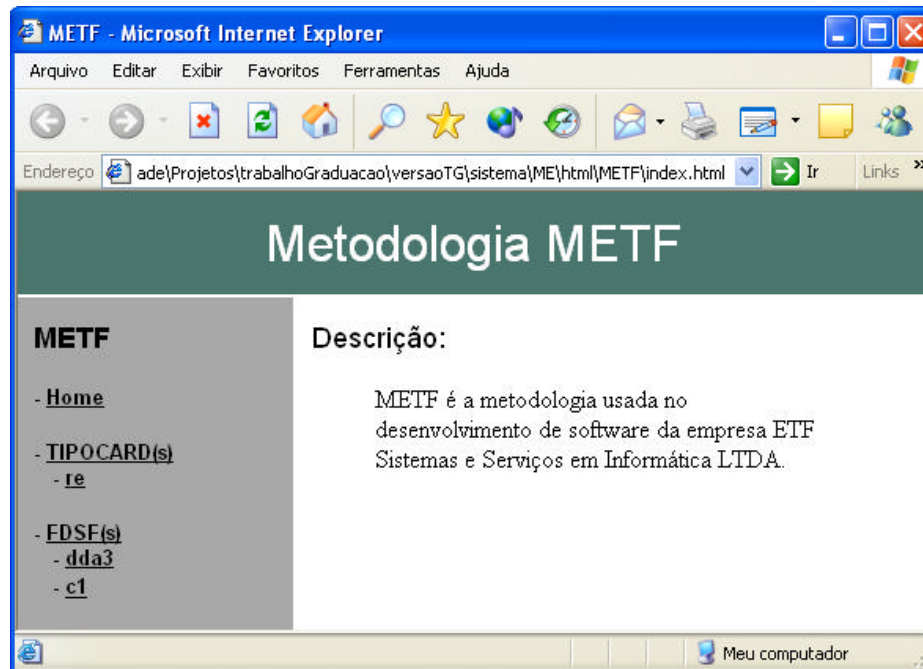
**Figura 24.** *Methodology Explorer Web Publisher*

A ferramenta só é aberta para metodologias já salvas. Portanto, antes de chamar o *Web Publisher* pelo menu é necessário salvar a metodologia de trabalho. Uma vez aberta, no passo 1, a ferramenta apresenta campos para escolha do diretório onde será gerado o *site* e o nome padrão para o arquivo inicial deste *site*. O passo-a-passo já pode ser finalizado neste ponto. Caso o usuário deseje, pode seguir ao passo 2, onde terá a opção para desmarcar a opção de exportação dos arquivos. Por *default*, os *sites* são gerados contendo *links* para os arquivos que compõem os conteúdos de componentes e que são exportados para subdiretórios dentro do diretório selecionado pelo usuário.



**Figura 25.** *Passos do Web Publisher*

No diretório selecionado pelo usuário será gerado um diretório de mesmo nome do título da metodologia e, dentro deste, serão gerados todos os arquivos HTML, subdiretórios auxiliares e os arquivos anexados a cada componente. A Figura 26 traz um exemplo de *site* gerado pelo *Web Publisher* para a metodologia “METF”.



**Figura 26.** Exemplo de *site* gerado pelo *Methodology Explorer Web Publisher*

## 8 Conclusões

Este capítulo finaliza o trabalho através da apresentação de suas principais conclusões. Nas seções seguintes são abordados os resultados que foram alcançados, as principais dificuldades encontradas na sua realização e possíveis melhorias que podem ser feitas como trabalhos futuros.

### 8.1 Resultados Alcançados

Concluimos o trabalho com a satisfação de ter o dever cumprido. As metas traçadas para o projeto foram seguidas à risca e todo o planejamento foi realizado a contento. Hoje deixamos um Methodology Explorer totalmente remodelado, mais poderoso, fácil e agradável de usar.

A remodelagem da interface gráfica consumiu um esforço significativo iniciado com um processo de avaliação de usabilidade, utilizando métodos empíricos de avaliação [7]. Estes foram de fundamental importância para a reestruturação da implementação da interface, se adequando mais às necessidades dos usuários. Hoje temos uma interface mais bem diagramada, privilegiando separação conceitual de elementos e mapeamento explícito de controle gráficos.

A ferramenta foi expandida com a criação de novas funcionalidades, desde o início consideradas essenciais para os propósitos que a ferramenta deveria atender. Exemplos delas são os novos recursos de publicação de metodologias e análise comparativa de metodologias.

Além da criação de novos recursos, algumas outras funcionalidades já existentes foram reprojatadas para se tornarem mais poderosas ou mais eficazes na realização de seus objetivos. A criação dos componentes já permitia definição de anexos através de *links* para URLs ou para o sistema de arquivos. Agora já é possível inserir o próprio arquivo diretamente na ferramenta. Outras funcionalidades, como a criação do mecanismo de instanciação de metodologias, foram refeitas simplesmente para se adequarem melhor aos novos padrões de usabilidade da nova interface.

Todas essas mudanças e evoluções passaram também por uma profunda reestruturação interna da ferramenta. Foi preciso refazer a estrutura de classes e relacionamentos entre elas, além de remodelar todo o banco de dados. Como resultado disso, as

funcionalidades prometidas puderam ser mais facilmente implementadas. Mais importante ainda, e talvez o resultado mais importante do projeto, foi que a reestruturação da arquitetura possibilitou que o Methodology Explorer deixasse de ser uma ferramenta baseada em instâncias do RUP para se tornar uma ferramenta genérica, capaz de ser usada para definir metodologias independentes de qualquer outra já criada. Essa característica juntamente com a possibilidade de reuso de componentes se constituem na principal utilidade do Methodology Explorer.

## 8.2 Dificuldades Encontradas

Apesar de todo o cronograma ter sido cumprido e os resultados todos alcançados, a execução do trabalho foi dificultada por alguns fatores referentes à complexidade dos conceitos envolvidos com a ferramenta.

O fato de existir a necessidade constante de manipulação de estruturas hierárquicas provocou uma série de erros durante a implementação. O elemento básico tratado era uma metodologia, que por sua vez, era composta por componentes e estes da mesma forma. Falando num nível mais de codificação, a manipulação desses elementos exigia que uma grande quantidade de linhas código fosse criada antes que qualquer teste pudesse ser feito. Além disso, as estruturas hierárquicas normalmente sugerem uso de estruturas de árvores. Estas são historicamente conhecidas como uma das estruturas de dados mais difíceis, ou pelo mais complexas, de manipulação de toda computação. Junto com o estudo de árvores, seguem diversos algoritmos de manipulação e busca em grafos, que tiveram que ser incorporados à implementação do Methodology Explorer. Foi freqüente o uso de funções recursivas para implementação desses algoritmos.

Como estamos tratando de uma ferramenta gráfica, uma dificuldade a mais foi ter que também trabalhar com árvores graficamente. Além da estrutura de objetos hierárquica criada, foi criada também uma estrutura similar através do uso das estruturas de árvores gráficas da API de Swing de Java. A classe principal utilizada foi a `JTree`, do pacote `javax.swing`. A manipulação de `JTree` requer uso de uma série de classes auxiliares, mas nem por isso dispensáveis, do pacote `javax.swing.tree`, que possui as classes e interfaces para manipular nós e ramos de uma `JTree`.

O uso de `JTree` gerou a necessidade de manutenção de consistência entre a árvore gráfica, apresentada ao usuário, e a árvore de componentes, que era manipulada internamente pela ferramenta. Pensou-se na hipótese de utilização de apenas o `JTree` nos dois casos, porém a idéia foi logo descartada devido a maior dificuldade que isso traria. `JTree` é muito boa, mas foi feita apenas para uso em interfaces gráficas. Sendo assim alguns algoritmos tiveram que ser criados para possibilitar conversão em dois

sentidos: de árvore de componentes para árvores *JTree* e de *JTree* para árvores de componentes. Mais uma complexidade incluída.

Outra dificuldade surgiu através da manipulação de metodologias em memória. Esse recurso foi incluído para melhorar a performance do sistema e diminuir a quantidade de acessos à base de dados. Isso foi muito útil, mas também complexo de implementar. Durante o trabalho, o usuário pode abrir uma metodologia e realizar uma série de operações, como criar novos componentes, alterar alguns e remover outros. Essas operações são refletidas imediatamente na tela para o usuário, mas ficam guardadas em memória antes de serem confirmadas no banco de dados. Quando o usuário realmente confirma a operação, um mecanismo complexo deve entrar em ação para identificar as diferenças entre a árvore gráfica, alterada pelo usuário, e de componentes, que deverá refletir as alterações da árvore gráfica antes de ser refletida definitivamente no banco de dados.

### 8.3 Trabalhos Futuros

De início podem ser identificados alguns pontos para serem trabalhados no futuro, principalmente em relação aos três *wizards* criados.

O *wizard* de publicação hoje trabalha com a geração de *site* para metodologias. Seria interessante incluir facilidades para personalização dos *sites* gerados para permitir que o usuário possa definir melhor suas seções, padrões de cores, incluir figuras diretamente através do *wizard*. A idéia é deixar o *site* o mais coerente possível com o usuário em questão, para que ele tenha pouca ou nenhuma necessidade de alteração após ser gerado. Outra funcionalidade realizada a publicação HTML, seria a criação de um recurso que possibilitasse um *preview* do *site* sendo gerado. Assim, o usuário poderia realizar alterações antes de ser gerado o produto final. Também aumentaria muito a poder do gerador de *sites*, se estes pudessem conter figuras que representassem graficamente os componentes ordenados da metodologia, como, por exemplo, figuras dos fluxos de atividades do RUP.

Também seria desejável que outras possibilidades de formatos de publicação pudessem existir. Hoje apenas em HTML, quem sabe, no futuro, o usuário possa escolher entre uma lista de opções como DOC, PDF, HTML, LaTeX e RTF.

Dentro da ferramenta só é possível visualizar as metodologias no formato de árvore. A ferramenta ganharia muito com a possibilidade de visualização de mais informações. Por exemplo, a visualização dos arquivos texto ou HTML anexados. Além da visualização, também deveria ser permitido o usuário realizar alterações no conteúdo desses arquivos.



O *wizard Analyser*, hoje é usado para comparar metodologias baseadas nas regras existentes nos tipos presentes em cada uma. Outra forma de incrementá-lo poderia ser através da análise parcial das metodologias, como a análise de apenas um conjunto de componentes. A análise poderia verificar a adequação de alguns componentes às regras definidas para seus tipos. Para isso, seria preciso incrementar o conjunto de regras existentes hoje, com a definição de novas regras, e melhorar o mecanismo de avaliação das mesmas.

Muito foi feito, mas muito ainda pode ser melhorado no Methodology Explorer. Consideramos que esse foi o primeiro passo no sentido de torná-lo uma ferramenta realmente com ambições comerciais.

# Referências

- [1] Maranhão, S. M. B.; Moura, H. P. "Documento de Requisitos do Sistema Methodology Explorer". Versão Alfa. 09/2001. Disponível: *site* Methodology Explorer (nov. 2002) URL: <http://www.cin.ufpe.br/~mexplorer>.
- [2] Maranhão, S. M. B. "Relatório Final de Atividades do bolsista UFPE/CNPq/PIBIC". 10/2001. Disponível: *site* Methodology Explorer (nov. 2002) URL: <http://www.cin.ufpe.br/~mexplorer>.
- [3] Kruchten, P. "The Rational Unified Process: An Introduction". Addison-Wesley Pub Co; 2nd edition, 2000.
- [4] Maranhão, S. M. B.; Moura, H. P. "Methodology Explorer: uma ferramenta para definição de metodologias de desenvolvimento de software". Apresentação à Quali Software Processes. 10/2001. Disponível: *site* Methodology Explorer (nov. 2002) URL: <http://www.cin.ufpe.br/~mexplorer>.
- [5] Maranhão, S. M. B.; Moura, H. P. "Methodology Explorer: uma ferramenta para definição de metodologias de desenvolvimento de software". Anais do CONIC 2001.
- [6] Ebling, M; John, B. E. "On the Contributions of Different Empirical Data in Usability Testing". Symposium on Designing Interactive Systems 2000.
- [7] Silva Jr, C.R.; Valença, R. V. A. "Utilizando Métodos Empíricos para Melhorar a Usabilidade e de Interfaces Gráficas". 10/2002. Disponível: *site* Methodology Explorer (nov. 2002) URL: <http://www.cin.ufpe.br/~mexplorer>.
- [8] Norman, D. A. "The Psychology of Everyday Things", New York: Basics Books, 1988;
- [9] Silva Jr, C.R.; Moura, H. P. "Documento de Requisitos do Sistema Methodology Explorer". Versão 1.0. 02/2003. Disponível: *site* Methodology Explorer (nov. 2002) URL: <http://www.cin.ufpe.br/~mexplorer>.
- [10] QSR NUD\*IST. Disponível: *site* QSR NUD\*IST (set. 2002). URL: [http://wizard.ucr.edu/wizard\\_project/nudist.html](http://wizard.ucr.edu/wizard_project/nudist.html).
- [11] Apache XML Project. Disponível: *site* Apache XML Project (15 jan. 2003). URL: <http://xml.apache.org/>.