

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO E SISTEMAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE  
PRODUÇÃO**

**DESENVOLVIMENTO E IMPLEMENTAÇÃO DE UM PROTÓTIPO  
DE FERRAMENTA PARA REUTILIZAÇÃO DE PLANOS DE  
MENSURAÇÃO UTILIZANDO RACIOCÍNIO BASEADO EM  
CASOS**

Dissertação submetida à Universidade Federal de Santa Catarina para a  
obtenção do título de Mestre em Engenharia de Produção.

**Marcos Roberto Rodrigues**

**Florianópolis, Brasil**

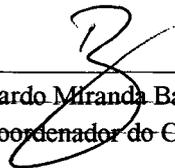
**Abril 2000**

**Marcos Roberto Rodrigues**

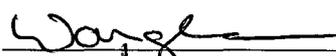
**Desenvolvimento e Implementação de um Protótipo de Ferramenta para Reutilização de Planos de Mensuração Utilizando Raciocínio Baseado em Casos**

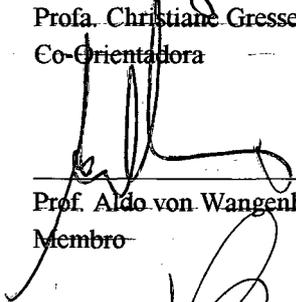
Esta dissertação foi julgada e aprovada para a obtenção do título de Mestre em Engenharia de Produção no Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina.

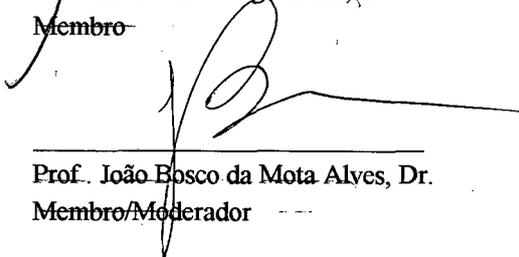
Florianópolis, 20 de Abril de 2000

  
\_\_\_\_\_  
Prof. Ricardo Miranda Barcia, Ph.D.  
Coordenador do Curso.

**Banca Examinadora:**

  
\_\_\_\_\_  
Prof. Christiane Gresse von Wangenheim, Dra.  
Co-Orientadora

  
\_\_\_\_\_  
Prof. Aldo von Wangenheim, Dr..  
Membro

  
\_\_\_\_\_  
Prof. João Bosco da Mota Alves, Dr.  
Membro/Moderador

## **Agradecimentos**

Ao professor Ricardo M. Barcia pela orientação, recursos e tempo dedicado para poder desenvolver e concluir este trabalho.

A professora Christiane Gresse von Wangenheim, pela paciência, em sua co-orientação, tempo e empenho dedicados no desenvolvimento e conclusão deste trabalho.

Aos Professores Membros da Banca, pela dedicação e os comentários e sugestões construtivos ajudando no melhoramento deste trabalho.

A minha esposa Simone e filha Sabrina, pela compreensão nos momentos de ausência, motivação e estímulo para a continuidade deste trabalho.

Aos amigos Harley Miguel Wagner, Herculano de Biasi e Paulo Roberto Delani, pelas discussões, auxílio, e companhia durante a execução deste trabalho.

E, por fim, a todas as pessoas que participaram, direta, ou indiretamente, na conclusão deste trabalho.

## **Resumo**

Uma infraestrutura essencial para o melhoramento do processo de software e o gerenciamento de projetos é a mensuração de software. Seu planejamento e implementação bem sucedida requer, na prática, uma quantidade significativa de esforço e experiência.

O Custo pode ser reduzido e a qualidade da mensuração pode ser melhorada reutilizando-se experiências adquiridas de programas de mensuração do passado.

Para a criação sistemática e a comunicação por toda a empresa do conhecimento de mensuração, experiências específicas da organização devem ser sistematicamente coletadas, armazenadas em memórias corporativas e reutilizadas em futuros projetos de software.

Entretanto, sistemas de aprendizagem baseados em conhecimento devem ser operacionalizados e integrados no processo de planejamento de mensuração.

Este trabalho apresenta uma abordagem customizável de raciocínio baseado em casos para a captura e reutilização de experiências de programas de mensuração de software.

Uma ferramenta, suportando todas as tarefas do processo de planejamento e possibilitando uma recuperação orientada a metas, e um mecanismo de recuperação baseado em similaridade, e aprendizagem contínua é apresentada.

## **Abstract**

One essential infrastructure for software process improvement and project management is software measurement. Its planning and successful implementation requires, in practice, a significant amount of effort and expertise.

Cost may be reduced and quality of measurement may be improved reusing experiences gathered from past measurement programs.

For the systematic creation and company-wide communication of measurement know-how, organization-specific experiences have to be collected systematically, stored in corporate memories and reused in future software projects.

Therefore, learning knowledge-based systems have to be operationalized and integrated into the measurement planning process.

This work presents a tailorable case-based reasoning approach for capturing and reusing experiences on software measurement programs.

A tool supporting all tasks of the measurement planning process and enabling the goal-oriented and similarity-based retrieval and continuous learning of measurement know-how is presented.

## SUMÁRIO

<b>SUMÁRIO .....</b>	<b>6</b>
----------------------	----------

<b>ÍNDICE DE FIGURAS.....</b>	<b>10</b>
-------------------------------	-----------

<b>ÍNDICE DE ABREVIATURAS.....</b>	<b>12</b>
------------------------------------	-----------

<b>INTRODUÇÃO .....</b>	<b>13</b>
-------------------------	-----------

<b>1 MOTIVAÇÃO .....</b>	<b>13</b>
--------------------------	-----------

<b>2 OBJETIVOS .....</b>	<b>18</b>
--------------------------	-----------

2.1 OBJETIVO GERAL .....	18
--------------------------	----

2.2 OBJETIVOS ESPECÍFICOS .....	18
---------------------------------	----

<b>3 ESTRUTURA DO TRABALHO .....</b>	<b>19</b>
--------------------------------------	-----------

<b>FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>20</b>
------------------------------------	-----------

<b>1. O PARADIGMA GOAL/QUESTION/METRICS.....</b>	<b>20</b>
--	-----------

1.1 O PROCESSO GQM .....	21
--------------------------	----

1.2 PROCESSO DE PLANEJAMENTO DE PROGRAMA DE MENSURACAO BASEADO EM GQM .....	22
---	----

1.2.1 Estudo Prévio .....	22
---------------------------	----

1.2.2 Identificação das Metas GQM .....	22
1.2.3 Desenvolvimento do plano GQM .....	23
1.2.4 Formalização da Meta .....	23
1.2.5 Desenvolvimento do nível de questões .....	25
1.2.6 Desenvolvimento dos modelos de qualidade .....	25
1.2.7 Definição das medidas .....	26
1.3 DESENVOLVIMENTO DO PLANO DE MENSURAÇÃO .....	26
1.3.1 Definição dos procedimentos de mensuração .....	26
1.3.2 Definição dos instrumentos de coleta de dados .....	27
<b>2. A FÁBRICA DE EXPERIÊNCIAS .....</b>	<b>27</b>
2.1 PARADIGMA DE MELHORAMENTO DE QUALIDADE .....	27
2.2 A FÁBRICA DE EXPERIÊNCIAS .....	29
<b>3. RACIOCÍNIO BASEADO EM CASOS .....</b>	<b>31</b>
3.1 FUNDAMENTOS DO RACIOCÍNIO BASEADO EM CASOS .....	32
3.1.1 Representação de Casos .....	33
3.1.2 O Modelo de Memória Dinâmica .....	33
3.1.3 O modelo Categoria & Exemplar .....	34
3.1.4 Indexação .....	35
3.1.5 Armazenamento .....	35
3.1.6 Recuperação .....	36
3.1.7 Adaptação .....	37
3.1.8 Aprendizado .....	38
<b><u>ANÁLISE COMPARATIVA DAS FERRAMENTAS EXISTENTES .....</u></b>	<b><u>39</u></b>
<b>1. GQMASPECT .....</b>	<b>39</b>
<b>2. GQM-TOOL .....</b>	<b>42</b>
<b>3. ES-TAME .....</b>	<b>43</b>
<b>4. GQM-DIVA .....</b>	<b>44</b>
<b>5. GQM-PLAN .....</b>	<b>45</b>
<b>6. COMPARAÇÃO ENTRE AS FERRAMENTAS APRESENTADAS .....</b>	<b>46</b>
<b><u>ABORDAGEM BAEADA EM CASOS PARA REUTILIZAÇÃO DE EXPERIÊNCIAS DE</u></b>	
<b><u>MENSURAÇÃO .....</u></b>	<b><u>48</u></b>

<b>1. REPRESENTAÇÃO DE EXPERIÊNCIAS DE MENSURAÇÃO .....</b>	<b>49</b>
<b>2.1. CASOS DE EXPERIÊNCIAS DE PRODUTOS GQM (GQM-PEC) .....</b>	<b>50</b>
<b>2.2. CASOS DE EXPERIÊNCIAS DE PROBLEMAS E SOLUÇÕES GQM (GQM-PSEC).....</b>	<b>51</b>
<b>2.3 CONHECIMENTO GERAL DO DOMÍNIO .....</b>	<b>51</b>
2.3.1 MODELOS DE PRODUTOS GQM .....	51
2.3.2 DEFINIÇÃO DE TIPOS .....	52
2.3.3 GLOSSÁRIOS .....	52
<b>4. SUPORTE BASEADO EM EXPERIÊNCIAS PARA PLANEJAMENTO GQM .....</b>	<b>52</b>
4.1 RECUPERANDO CASOS DE EXPERIÊNCIAS RELEVANTES .....	54
4.1.1 Manutenção das metas de recuperação.....	55
4.2 SUPORTE INTERATIVO PARA A REUTILIZAÇÃO DE CASOS DE EXPERIÊNCIAS.....	57
4.3 AQUISIÇÃO DO CONHECIMENTO ESPECÍFICO DO PROJETO .....	57
4.4 INTEGRAÇÃO DE NOVAS EXPERIÊNCIAS .....	59
<b>5. RECUPERAÇÃO DE EXPERIÊNCIAS DE ENGENHARIA DE SOFTWARE.....</b>	<b>60</b>
5.1 AVALIAÇÃO DA SITUAÇÃO .....	60
5.2 BUSCA EXATA DOS ÍNDICES MARCADOS COMO ESSENCIAIS .....	60
5.3 COMPARAÇÃO PARCIAL DOS CASOS .....	60
5.4 SELEÇÃO DOS CANDIDATOS A REUTILIZAÇÃO.....	61
5.4.1 Medida de similaridade para recuperação de experiências de engenharia de software.....	61
5.4.2 Medida de Similaridade para a Recuperação de experiências de Engenharia de Software.....	68
<b><u>A FERRAMENTA REMEX .....</u></b>	<b><u>70</u></b>
<b>1. EDIÇÃO/PLANEJAMENTO DE UM PROGRAMA DE MENSURAÇÃO .....</b>	<b>72</b>
1.1 CARACTERIZAÇÃO DO CONTEXTO .....	72
1.1.1 Caracterização da Organização.....	73
1.1.2 Caracterização do Projeto .....	74
1.1.3 Caracterização da Mensuração .....	75
1.2 DEFINIÇÃO DO PLANO DE MENSURAÇÃO .....	76
1.2.1 Identificação da Meta de Mensuração .....	76
1.2.2 Execução das Entrevistas .....	77
1.2.3 Definição das Perguntas GQM .....	78
1.2.4 Definição dos Modelos de Qualidade .....	79
1.2.5 Definição das Medidas GQM .....	80

1.3 DEFINIÇÃO DO PLANO DE MENSURAÇÃO.....	82
1.3.1 Procedimentos de Coleta de Dados.....	83
1.3.2 Instrumentos de Coleta de Dados.....	83
<b>2. RECUPERAÇÃO DE EXPERIÊNCIAS .....</b>	<b>84</b>
2.1 RECUPERAÇÃO DE PEC .....	84
2.2 RECUPERAÇÃO DE PSEC.....	85
2.2.1 Prevenção de Problemas .....	86
2.2.2 Resolução de Problemas .....	87
<b>3. AQUISIÇÃO DO CONHECIMENTO .....</b>	<b>88</b>
3.1 AQUISIÇÃO DE CASOS DE PEC'S.....	89
3.2 AQUISIÇÃO DE PSEC'S.....	90
<b>4. MANUTENÇÃO DO SISTEMA.....</b>	<b>90</b>
4.1 HIERARQUIA DE OBJETOS.....	90
4.2 HIERARQUIA DE TIPOS .....	91
4.3 ATRIBUTOS .....	92
4.4 METAS DE RECUPERAÇÃO.....	93
4.5 BASE DE CASOS .....	94
<b><u>CONCLUSÃO .....</u></b>	<b><u>96</u></b>
<b><u>REFERÊNCIAS BIBLIOGRÁFICAS.....</u></b>	<b><u>98</u></b>

## ÍNDICE DE FIGURAS

FIGURA 1. DIMENSÕES DA META GQM.	23
FIGURA 2. ABSTRACTION SHEET.	25
FIGURA 3. A FÁBRICA DE EXPERIÊNCIAS.	30
FIGURA 4. EDIÇÃO DO ABSTRACTION SHEET NO GQMACCESS	40
FIGURA 5. O PAPEL DO GQMASPECT NO PROCESSO GQM	41
FIGURA 6. A FERRAMENTA GQM-TOOL.	42
FIGURA 7. INTERFACE DO GQM-DIVA	45
FIGURA 8. COMPARAÇÃO ENTRE AS FERRAMENTAS.	46
FIGURA10. MODELO DE META DE RECUPERAÇÃO.	56
FIGURA 11. TELA PRINCIPAL DA FERRAMENTA REMEX	71
FIGURA 12. MODO PLANEJAMENTO: CARACTERIZAÇÃO DA ORGANIZAÇÃO	73
FIGURA 13. MODO PLANEJAMENTO: CARACTERIZAÇÃO DO PROJETO	74
FIGURA 14. MODO PLANEJAMENTO: CARACTERIZAÇÃO DA MENSURAÇÃO	75
FIGURA 15. MODO PLANEJAMENTO: DEFINIÇÃO DA META DE MENSURAÇÃO	76
FIGURA 16. MODO PLANEJAMENTO: DEFINIÇÃO DO ABSTRACTION SHEET.	77
FIGURA 17. MODO PLANEJAMENTO: EDIÇÃO DOS ITENS DO ABSTRACTION SHEET	78
FIGURA 18. MODO PLANEJAMENTO: DEFINIÇÃO DAS PERGUNTAS	78
FIGURA 19. MODO PLANEJAMENTO: EDIÇÃO DE PERGUNTAS	79
FIGURA 20. MODO PLANEJAMENTO: DEFINIÇÃO DOS MODELOS DE QUALIDADE	79
FIGURA 21. MODO PLANEJAMENTO: EDIÇÃO DE MODELOS DE QUALIDADE	80
FIGURA 22. MODO PLANEJAMENTO: DEFINIÇÃO DAS MEDIDAS.	81
FIGURA 23. MODO PLANEJAMENTO: EDIÇÃO DE MEDIDAS	82
FIGURA 24. MODO PLANEJAMENTO: DEFINIÇÃO DOS PROCEDIMENTOS DE MENSURAÇÃO	83

FIGURA 25. MODO PLANEJAMENTO: DEFINIÇÃO DOS INSTRUMENTOS DE MENSURAÇÃO _____	84
FIGURA 26. RECUPERAÇÃO DE EXPERIÊNCIAS: MODELOS DE QUALIDADE. _____	85
FIGURA 27. RECUPERAÇÃO DE EXPERIÊNCIAS: PREVENÇÃO DE PROBLEMAS _____	86
FIGURA 28. RECUPERAÇÃO DE EXPERIÊNCIAS: DESCRIÇÃO DE UM NOVO PROBLEMA _____	87
FIGURA 29. RECUPERAÇÃO DE EXPERIÊNCIAS: RECUPERAÇÃO DE SOLUÇÕES _____	88
FIGURA 30. AQUISIÇÃO DE CONHECIMENTO: FILA DE CASOS A SEREM ADICIONADOS NA GQM-EB _____	89
FIGURA 31. MANUTENÇÃO DO SISTEMA: HIERARQUIA DE OBJETOS _____	91
FIGURA 32. MANUTENÇÃO DO SISTEMA: DEFINIÇÃO DE TIPOS _____	92
FIGURA 33. MANUTENÇÃO DO SISTEMA: DEFINIÇÃO DE ATRIBUTOS _____	93
FIGURA 34. MANUTENÇÃO DO SISTEMA: DEFINIÇÃO DAS METAS DE RECUPERAÇÃO _____	94
FIGURA 35. MANUTENÇÃO DO SISTEMA: BASE DE CASOS _____	95

## ÍNDICE DE ABREVIATURAS

GQM	Goal/Question/Metric
FE	Fábrica de Experiências
EW	Experienceware

## INTRODUÇÃO

### 1 Motivação

Hoje, quase todo negócio envolve o desenvolvimento ou uso de software. Porém o software é tampouco o principal aspecto do negócio, o valor adicionado ao produto ou serviço está no caminho crítico para o sucesso.

Assim como organizações estão tornando-se mais dependentes de software, a significância da qualidade e produtividade dos negócios de software aumenta e torna-se de importância essencial para a competitividade de uma organização.

Entretanto, o estado da prática é que sistemas de software freqüentemente carecem de qualidade e muitos projetos de software que não cumprem o prazo planejado e com orçamentos ultrapassados.

Desenvolvimento e manutenção de software praticamente não tem acompanhado a crescente demanda por qualidade de software, o aumento da complexidade e instabilidade dos sistemas de software e os rápidos avanços tecnológicos.

Em muitas organizações os processos de desenvolvimento e manutenção de software foram estabelecidos de maneira informal, o que freqüentemente leva à elaboração de soluções incompatíveis com as reais necessidades.

Para a melhoria contínua da qualidade e produtividade de software, é necessário

entender os produtos e processos de software de uma organização em particular. Isto requer que essa organização construa seu próprio conhecimento de Engenharia de Software (por ex. modelos de custo, distribuição de defeitos) e compartilhe este conhecimento entre seus projetos de software. Neste contexto, o *Quality Improvement Paradigm* (QIP) [BCR94a] é uma abordagem para o melhoramento contínuo e a execução sistemática de projetos de software.

É um processo interativo o qual inclui o planejamento, execução e avaliação dos melhoramentos, assim como a incorporação da experiência adquirida a partir dos esforços de melhoramento para futuros projetos. Um pré-requisito básico é o entendimento do processo atual na organização e a identificação de forças e fraquezas dos processos e produtos específicos.

Para alcançar este entendimento, programas de mensuração devem ser estabelecidos [BCR94a]. Somente a mensuração provê dados quantitativos e qualitativos sobre projetos de software, processos e produtos, levando em consideração as características específicas da empresa e metas de negócios, para suportar:

- Entendimento dos processos e produtos de software específicos da organização (por ex., distribuição do esforço por fases ou percentual de falhas detectadas por inspeções),
- Isolamento de fatores de contexto que têm um impacto sobre a qualidade e a produtividade (por ex. linguagens de programação, experiência dos desenvolvedores ou plataforma de desenvolvimento),
- Identificação de forças ou fraquezas de produtos e processos (por ex., gargalos entre fases ou tipos de defeitos por módulos), e;
- Monitoramento dos resultados das modificações efetuadas com o objetivo de melhoramento (por ex. se a introdução de inspeções do projeto de fato reduz o número de defeitos).

Neste contexto, o paradigma *Goal/Question/Metric* (GQM) [BCR94b] é uma

tecnologia avançada para a mensuração orientada a metas em projetos de software, suportando a definição e implementação de metas de melhoramento de software mensuráveis e operacionalizáveis.

Para ser efetivo e eficiente, programas de mensuração devem ser ajustados às características específicas da organização, seus processos de software, produtos e metas de negócio. Portanto um dos maiores fatores de sucesso de um programa de mensuração é o seu planejamento apropriado.

Uma vez que isto é um processo intelectual complexo, o qual requer pessoas experientes, é normalmente custoso e a probabilidade de se cometerem erros é alta. Por essa razão, a disponibilidade do conhecimento explícito de mensuração, ajustado às características específicas da organização e suas necessidades, podem contribuir significativamente para o melhoramento do planejamento de programas de mensuração e levar a uma substancial redução do esforço.

Isso possibilita a consolidação do conhecimento de mensuração por toda a companhia em competências que fortalecem a empresa para alcançar consideráveis melhoramentos e benefícios.

Em geral, vários tipos de *experienceware* (EW) [Gre98], incluindo especialidades e lições aprendidas (por ex. como motivar a coleta de dados), modelos de qualidade (por ex. distribuição do esforço de re-trabalho por tipo de falha), ou qualquer produto desenvolvido durante um programa GQM (por ex. instrumentos de coleta de dados) são potencialmente reutilizáveis.

Embora haja freqüentemente uma necessidade de ajustar estes produtos para as características específicas do projeto, eles são largamente reutilizáveis.

Reutilizar produtos ou modelos de qualidade os quais foram desenvolvidos no mesmo contexto organizacional requer menos esforço e direciona-se mais às necessidades específicas.

Reutilizando lições aprendidas auxilia no melhoramento do planejamento de programas de mensuração, continuamente prevenindo a repetição de falhas e guiando a

efetiva e eficiente solução de problemas.

A disponibilidade de experiências de mensuração também promove a criação de competências de mensuração facilitando a disseminação ampla da aplicação de programas de mensuração na prática.

Hoje, na maioria dos casos, a disseminação por toda a empresa de produtos e lições aprendidas não é suportada. A reutilização de conhecimento de mensuração é feita *ad-hoc*, de maneira informal, normalmente limitada às experiências pessoais.

Para maximizar a produtividade e os ganhos de qualidade pela reutilização de experiências, atividades de reutilização devem ser sistematizadas pelo acompanhamento de uma definição de processo de gerenciamento de conhecimento e suportada por uma infraestrutura técnica e organizacional, buscando a capitalização e reutilização de experiências por toda a companhia.

Então, Sistemas de Gerenciamento de Memórias Corporativas para a aquisição sistemática, a armazenagem, em memórias corporativas, e comunicação destas experiências devem ser construídas, operacionalizando o «aprendizado a partir de experiências» em ambientes industriais.

Como uma estrutura física e lógica de memórias corporativas para a contínua construção do conhecimento de software em uma organização, a abordagem de Fábrica de Experiência (FE) [BCR94a] tem se revelado como uma bem sucedida solução.

A abordagem de FE propõe uma infraestrutura para a análise e síntese de todos os tipos de experiências, atuando como um repositório para estas, e suprindo estas experiências para projetos em demanda. Isto ainda é complicado pela complexidade dos programas de mensuração, o que torna o entendimento e a identificação de experiências de mensuração relevantes e reutilizáveis mais difícil.

Para suportar compreensivelmente o processo de planejamento de mensuração, vários tipos de experiências relacionados a diversas tarefas (por ex. desenvolvimento de medidas ou de instrumentos de coleta de dados) em diferentes ambientes têm sido recuperados dirigindo-se a vários propósitos: facilitação do desenvolvimento de produtos de

mensuração, prevenção de falhas do passado pela antecipação de problemas, e guia na solução de problemas que estão ocorrendo.

E, uma vez que cada projeto de software é diferente, é muito difícil encontrar um artefato que preencha completamente as necessidades do atual situação, ainda complicado pela falta de conhecimento geral de mensuração na prática.

Então, é necessária a construção de uma infraestrutura técnica, que possibilite uma recuperação inteligente e ajustável a diferentes objetivos e contextos de reutilização, e a sistemática e contínua aquisição de novas experiências.

Para atingir esta meta, um sistema baseado em conhecimento capaz de aprender deve ser construído como uma plataforma integrada de suporte, operacionalizando a abordagem da Fábrica de Experiências<sup>1</sup>.

Neste contexto, o *Raciocínio Baseado em Casos* (RBC) [AP94] tem um papel chave, uma vez que provê um pleno suporte para o desenvolvimento de sistemas baseados em conhecimento.

As maiores vantagens do RBC neste contexto são a recuperação baseada em similaridade para todos os tipos de artefatos e seu foco primário em conhecimento baseado em experiência.

A aprendizagem contínua e incremental ocorre naturalmente através do produto da resolução de problemas e da revisão e captura de experiências cada vez que um novo problema é resolvido.

---

<sup>1</sup> Neste trabalho, não estamos considerando os fatores gerenciais da abordagem.

## 2 Objetivos

### 2.1 Objetivo Geral

Desenvolver e Implementar o protótipo de uma ferramenta que suporte o planejamento de planos de programas de mensuração baseados no paradigma Goal/Question/Metric (GQM), possibilitando a reutilização baseada em metas de experiências adquiridas em planejamentos de mensuração passados, incluindo lições aprendidas em problemas e soluções aplicadas e produtos GQM, suportando assim o aprendizado organizacional.

### 2.2 Objetivos Específicos

Estudar o paradigma Goal/Question/Metrics para utiliza-lo como metodologia para a elaboração de planos de mensuração utilizados no processo de melhoria de qualidade de processos e produtos de software.

Estudar os conceitos de implementação de uma Fábrica de Experiências (FE) tornando-a efetiva e eficaz no armazenamento e evolução incremental da base de experiências de mensuração obtida a partir dos programas de mensuração efetuados.

Estudar o uso de Raciocínio Baseado em Casos em uma abordagem que forneça métodos para recuperação orientada a metas e para contínua aquisição e integração de novas experiências levando em consideração as características específicas do domínio

Efetuar um rápido comparativo das ferramentas existentes que suportam o planejamento de programas de mensuração baseados no paradigma GQM, no que diz respeito à abrangência destas nas fases de planejamento GQM, e assistência aos engenheiros de qualidade na implementação dos respectivos planos de mensuração.

Apresentar uma abordagem baseada em casos para a implementação de uma plataforma de suporte ao aprendizado organizacional de experiências de mensuração

ajustáveis às necessidades da organização.

Apresentar um protótipo da ferramenta desenvolvida a partir da abordagem sugerida, suas características e funcionalidade.

### 3 Estrutura do Trabalho

Este trabalho está dividido em quatro capítulos assim distribuídos:

No primeiro capítulo apresenta uma breve revisão teórica a respeito dos principais técnicas utilizadas na construção da ferramenta, dentre as quais estão o Paradigma Goal/Question/Metric (GQM), o conceito de Fábrica de Experiências e a técnica de Raciocínio Baseado em Casos.

O segundo capítulo apresenta uma descrição das ferramentas existentes que auxiliam no planejamento de mensuração baseada em GQM, efetuando uma comparação entre suas principais características.

O terceiro capítulo demonstra a abordagem técnica usada para a implementação da ferramenta, apresentando os conceitos necessários à sua fundamentação.

O quarto capítulo apresenta as principais características da ferramenta desenvolvida, demonstrando as facilidades de planejamento de programas de mensuração baseado em GQM, recuperação de experiências e aquisição do conhecimento obtido durante o processo de planejamento.

## FUNDAMENTAÇÃO TEÓRICA

### 1. O PARADIGMA GOAL/QUESTION/METRICS

A mensuração é uma técnica essencial na busca pelo melhoramento sistemático do processo de software. A mensuração provê dados quantitativos e qualitativos de projetos de software, processos e produtos o qual suporta, de maneira a:

1. Entender processos e produtos específicos da organização;
2. Isolar fatores de contexto com impacto na qualidade e produtividade do software;
3. Identificar forças e fraquezas de produtos e processos, e;
4. Acompanhar resultados de modificações feitas, na tentativa de atingir melhoramentos.

Neste contexto, o paradigma Goal/Question/Metric (GQM) [GRR98,BDR96,BCR94b,BW84] é uma ótima tecnologia para mensuração orientada a metas em projetos de software, suportando a definição e implementação de metas de melhoramento, operacionais e mensuráveis. GQM representa a mais avançada abordagem para mensuração orientada a metas, assim como provê um guia preciso para a formulação e metas de mensuração, a derivação top-down das medidas e a análise subsequente e interpretação dos dados de mensuração. Em comparação à abordagem Software Quality Metrics (SQM) [Boe81] e a abordagem Quality Function Deployment (QFD) [KA83] como duas abordagens diferentes de mensuração orientada a metas, a abordagem GQM suporta a análise de diferentes objetos, de diferentes pontos de vista, para diferentes propósitos.

Seguindo a abordagem GQM, a tarefa de análise da mensuração é especificada explicitamente pelo detalhamento da meta de mensuração. A medida é derivada de uma forma top-down baseada na meta de mensuração através de questões e modelos. Este refinamento é precisamente documentado em um plano GQM, provendo um funcionamento lógico para a seleção das medidas destacadas. A coleta de dados é interpretada de uma forma bottom-up no contexto da meta de mensuração, questões e modelos, considerando as limitações e suposições subjacentes a cada medida. A abordagem GQM foi aplicada com sucesso em diversas companhias como NASA-SEL [BCR94b,NAS94,BW84], Robert Bosch [BDT96], Digital [FLM+96], Motorola [Das92], SIA [CFG+98] e Schlumberger [LSO+98,HOR+96,SLO95].

### 1.1 O PROCESSO GQM

O método GQM [GRR98, BDR96] descreve o planejamento e a execução do processo de um programa de mensuração baseado no paradigma GQM, e suas principais fases são:

- **Estudo Prévio:** No início do programa de mensuração um pré-estudo é executado para identificar as metas de negócio e de melhoramento da organização e para caracterizar o ambiente específico. Um projeto piloto para a introdução do programa de mensuração é selecionado e caracterizado. Pré-requisitos necessários para estabelecimento da mensuração são completados. Pessoas envolvidas no programa de mensuração são treinadas e motivadas.
- **Identificação das metas GQM e desenvolvimento de planos GQM:** Baseado na caracterização do ambiente de mensuração, as metas do programa de mensuração são explicitamente definidas em termos de objeto, propósito, enfoque da qualidade, ponto de vista e contexto. As metas de mensuração são refinadas em um conjunto de medidas relevantes através de questões e modelos de qualidade, resultando em um plano GQM.
- **Desenvolvimento do plano de mensuração:** Um plano de mensuração é desenvolvido integrando as medidas dos planos GQM no processo de desenvolvimento do projeto de software estudado. Para as medidas definidas nos planos GQM, procedimentos

de mensuração são desenvolvidos, definindo para cada medida, quando, como e por quem os dados necessários serão coletados em relação a esses procedimentos, instrumentos de coleta de dados são desenvolvidos.

- Coleta de dados, análise e interpretação: Durante a fase de execução do programa de mensuração os dados são coletados de acordo com os procedimentos de coleta de dados especificados no plano de mensuração. Os dados coletados são validados e armazenados. Seguindo o plano GQM de baixo para cima, os dados de mensuração coletados são analisados e interpretados em *feedback sessions* envolvendo os participantes do projeto de software. Baseado nos conhecimentos ganhos, ações de melhoramento são decididas, e

- Análise Post-mortem: Os resultados da mensuração, incluindo os dados coletados e suas interpretações são analisadas após a finalização do projeto. Conhecimento ganho através da mensuração é formulada em relação aos fatores relevantes do contexto.

- Empacotamento: Os resultados da análise, conclusões e o conhecimento adquirido são empacotados em modelos de normas ou guias, etc.

## 1.2 PROCESSO DE PLANEJAMENTO DE PROGRAMA DE MENSURACAO BASEADO EM GQM

Para este trabalho nós enfocamos apenas o processo de planejamento de programas de mensuração os quais são descritos em detalhes a seguir [Gre00, GRR98]:

### 1.2.1 Estudo Prévio

### 1.2.2 Identificação das Metas GQM

Quando planejando um programa de mensuração orientado a metas, o primeiro passo é especificar as metas a serem alcançadas pelo programa de mensuração. Dependendo das metas de negócio e de melhoramento da organização e problemas existentes no processo de software, metas de mensuração potenciais devem ser identificadas com grande cuidado pelos engenheiros de garantia de qualidade.

Com base em uma descrição informal das metas de mensuração, as dimensões (ver figura 1) das metas GQM devem ser derivadas.

Objeto	Descreve o que será analisado. Em mensuração de Software, os objetos de interesse podem ser classificados em: <ul style="list-style-type: none"> <li>• Processos: São quaisquer atividades relacionadas com o software</li> <li>• Produtos: São quaisquer artefatos, ou documentos produzidos durante o os processos.</li> <li>• Recursos: São os itens de entrada dos processos</li> </ul>
Propósito	Descreve porque o objeto será analisado.
Foco de Qualidade	Descreve qual propriedade do objeto será analisado.
Ponto de Vista	Descreve quem irá utilizar os dados coletados.
Contexto	Descreve em que ambiente a análise é feita.

FIGURA 1. DIMENSÕES DA META GQM.

Isto deve ser realizado com cuidado, considerando todas as características ambientais, padrões, terminologias e como as conseqüências desta atividade podem ser dispendiosas de tempo.

### 1.2.3 Desenvolvimento do plano GQM

O objetivo do processo de desenvolvimento do plano GQM é operacionalizar as Metas GQM através de questões GQM, modelos e um conjunto de medidas GQM, definindo itens mensuráveis. Este processo é executado para cada meta GQM e resulta em um plano GQM definido para cada meta GQM.

### 1.2.4 Formalização da Meta

Para refinar a meta em medidas operacionais, o foco da qualidade necessita ser adequadamente definido e preencher as necessidades do ponto de vista, e também o objeto a ser analisado. O Conhecimento relevante é adquirido entrevistando as pessoas que representam o ponto de vista da meta GQM.

#### **1.2.4.1 Aquisição do conhecimento**

A informação adquirida através das entrevistas realizadas com os papéis envolvidos no ponto de vista da definição da meta GQM, e é utilizada para derivar modelos de qualidade válidos e corretos, para identificar fatores relevantes no contexto e então medidas relevantes. Os seguintes tópicos devem ser cobertos:

●Fatores de qualidade: Definem o conjunto de dimensões esperadas de qualidade, do ponto de vista representado na meta.

●Hipóteses base: Para cada dimensão de qualidade pertinente para o foco de qualidade, e a distribuição esperada dos valores podem ser definidas com base na intuição dos entrevistados e no conhecimento disponível do domínio.

●Fatores de variação: São definidos os fatores os quais se espera que tenham impacto nas dimensões de qualidade.

●Impacto nas hipóteses base: Para cada fator de variação, o impacto esperado do fator de variação na dimensão de qualidade pode ser especificado, quando possível.

Um instrumento normalmente usado na aquisição e estruturação do conhecimento durante as entrevistas é o *abstraction sheet* (ver figura 2). Um *abstraction sheet* é um documento de uma página com 4 quadrantes, um para cada um dos fatores acima descritos, com a respectiva meta GQM no cabeçalho.

Objeto	Propósito	Foco de Qualidade	Ponto de Vista	Ambiente
Processo de Desenvolvimento	Analisar	Causas da Qualidade	Equipe de Desenvolvimento	Empresa ABC
Foco de Qualidade		Fatores de Variação		
<ul style="list-style-type: none"> <li>• Número total de falhas.</li> <li>• Classificação das falhas por gravidade.</li> </ul>		<ul style="list-style-type: none"> <li>• Experiência da equipe de desenvolvimento</li> </ul>		
Hipóteses		Impactos		
<ul style="list-style-type: none"> <li>• Número total de falhas: 90</li> <li>• Distribuição por gravidade: 15% Alta, 60% Média, 25% Baixa.</li> </ul>		<ul style="list-style-type: none"> <li>• Equipe com mais experiência introduz menos falhas.</li> </ul>		

FIGURA 2. ABSTRACTION SHEET.

### 1.2.5 Desenvolvimento do nível de questões

O plano GQM é desenvolvido com base no conhecimento adquirido durante as entrevistas. Para cada dimensão da qualidade e fator de variação documentado no *abstraction sheet*, uma questão no plano GQM é refinada, expressando a necessidade por informação. As questões representam um refinamento intuitivo da meta de mensuração.

A hipótese pode visualizar uma questão e ser explicitamente baseada na hipótese dada no *abstraction sheet*. A hipótese especifica a distribuição esperada dos valores através das dimensões da qualidade ou fatores de qualidade focadas na questão no ambiente específico.

### 1.2.6 Desenvolvimento dos modelos de qualidade

Cada questão do plano GQM é formalizada pela definição detalhada de modelos de qualidade. Estes modelos provêm um significado para responder às questões.

Os Modelos operacionalizam as questões do plano GQM, quantificando os vários atributos abstratos dos artefatos estudados, e definindo precisamente como comparações de qualidade e produtividade, avaliações e predições são realizadas.

Entretanto características, padrões, e terminologias devem ser bem conhecidas. Os modelos são assim desenvolvidos abrangendo conceitos abstratos, como tamanho, com os quais serão refinados em modelos descritivos operacionais, para a mensuração.

Os Modelos devem ser cuidadosamente analisados para determinar a validade de suas hipóteses e sua aplicabilidade no ambiente em particular sob estudo. Com base na definição dos modelos, atributos relevantes serão medidos para que os modelos sejam determinados.

### 1.2.7 Definição das medidas

As questões do plano GQM são refinadas quantitativamente em um conjunto de medidas através de modelos de qualidade. Para cada atributo uma medida é definida. O nível de mensuração e a unidade e faixa dos respectivos dados a serem coletados também devem ser definidos.

## 1.3 Desenvolvimento do plano de mensuração

O foco principal do plano de mensuração é a integração apropriada das medidas no processo de desenvolvimento de software quando aplicadas no projeto de software atual.

### 1.3.1 Definição dos procedimentos de mensuração

Os procedimentos de coleta de dados são definidos pela determinação de cada medida identificação nos planos GQM de quem, como e quando os dados serão coletados. Em detalhe as seguintes atividades serão realizadas.

O primeiro passo, é a criação de uma lista única contendo todas as medidas a serem efetuadas, passando então a definição de quando, durante o processo de software, o dado requerido será coletado.

Três estratégias principais têm sido utilizadas: periodicamente; início/final das atividades/fases do processo de software; ou quando um artefato alcança um determinado estado.

Para cada medida, as pessoas ou ferramentas que tornam possível prover os dados serão identificadas. Os fatores de decisão são: se ele pode ser coletado automaticamente por ferramentas ou em qual papel ou posição na estrutura da organização é mais bem apresentada para a coleta de dados.

Devemos ainda, determinar quem é o responsável pela garantia da qualidade e armazenar ou manipular os dados.

### 1.3.2 Definição dos instrumentos de coleta de dados

O próximo passo é a especificação dos instrumentos de coleta de dados, definindo como os dados serão coletados para cada medida. Categorias principais de instrumentos de mensuração podem ser identificadas: ferramentas, questionários, e entrevistas estruturadas.

Para implementar o plano de mensuração, os instrumentos de coleta de dados requeridos serão desenvolvidos. Dependendo do tipo de instrumento de coleta de dados, alguma ferramenta será desenvolvida, questionários serão desenhados ou entrevistas estruturadas serão planejadas.

Para uma coleta confiável dos dados, as questões devem ser cuidadosamente formuladas, todos os termos desconhecidos explanados e as questões abertas limitadas, o quanto possível.

A usabilidade deve ser atingida para reduzir o esforço relacionado à coleta de dados, entretanto o questionário deve ser bem estruturado, colocando as questões em uma ordem lógica.

## 2. A FÁBRICA DE EXPERIÊNCIAS

### 2.1 Paradigma de Melhoramento de Qualidade

O Quality Improvement Paradigm [BCR94a] é a base metodológica para a construção da Fábrica de Experiências, tendo como base o Ciclo Plan/Do/Check/Act de Shewart-Deming [DEM86] largamente utilizado na indústria para a implementação de planos de gerenciamento de qualidade, sendo articulado nos seguintes passos:

1. Caracterizar: Entender o ambiente baseando-se nos modelos disponíveis, dados, intuição, etc. Estabelecer linhas mestras com o processo de negócio existente na organização e caracterizar sua severidade.

2. Definir Metas: As metas são definidas com base na caracterização inicial, na

forma de itens quantificáveis para melhoramento e da organização e projetos. Expectativas razoáveis são definidas com base nas linhas mestres, definidas na caracterização.

3. Escolher o Processo: Com base na caracterização do ambiente e das metas que foram definidas, escolher o processo apropriado para melhoramento, ferramentas e métodos de suporte, tendo certeza de que eles são apropriados com as metas que foram definidas.

4. Executar: Executar o processo, construindo os produtos e provendo retorno do projeto com base nos dados que são obtidos ao alcançar as metas.

5. Analisar: No final de cada projeto, analisar os dados e a informação adquirida para avaliar as práticas atuais, determinar problemas, buscar registros, e efetuar recomendações para futuros melhoramentos nos projetos.

6. Empacotamento: Consolidar a experiência adquirida na forma de novos, ou atualizados modelos e outras formas de conhecimento estruturado adquirido destes e outros projetos, e armazená-los em uma base de experiências para futuros projetos.

O Paradigma de Melhoria de Qualidade implementa dois ciclos de “*feedback*.”

- O ciclo do projeto: O “*feedback*” é provido para o projeto durante a fase de execução, que é provido para o projeto, durante a fase de execução, com o propósito de fornecer indicadores quantitativos ao nível de tarefa e projetos, a fim de prever e resolver problemas que venham a ocorrer.
- O ciclo da organização: O *feedback* é provido para a organização com o propósito de prover informações analíticas sobre a performance dos projetos, quando da conclusão dos mesmos. É utilizada para acumular experiências reusáveis que serão aplicáveis a outros projetos e são, em geral, melhorados com base na análise de desempenho.

Para executar o processo, experiências devem ser acessíveis em um formato empacotado como processos a serem escolhidos, produtos previamente desenvolvidos para reutilização, recursos apropriados e modelos de dados, e modelos de desenvolvimento de

software que permitam tirar vantagem dos pacotes reusáveis.

## 2.2 A Fábrica de Experiências

O Paradigma de Melhoramento de Qualidade está baseado na noção de que:

- O melhoramento de processos e produtos de software requer a acumulação contínua de experiências avaliadas (aprendizagem),
- De forma que podem ser efetivamente entendidas e modificadas (modelos de experiências),
- Em um repositório de modelos de experiências integrados (base de experiências),
- Que pode ser acessado e modificado para satisfazer as necessidades do projeto atual (reutilização).

O paradigma implica na separação lógica do desenvolvimento do projeto (executado pela organização de projeto) do aprendizado sistemático e empacotamento de experiências reutilizáveis.

A fábrica de experiências (FE) [BCR94a] é uma organização física e/ou lógica que suporta desenvolvimento de projetos pela análise e síntese de todos os tipos de experiências, agindo como um repositório para esta experiência, e suprindo a experiência para vários projetos em demanda.

Ela empacota experiências construindo modelos informais, formais ou esquematizados e medidas de vários processos e produtos de software, e outras formas de conhecimento de pessoas, documentos e suporte automatizado.

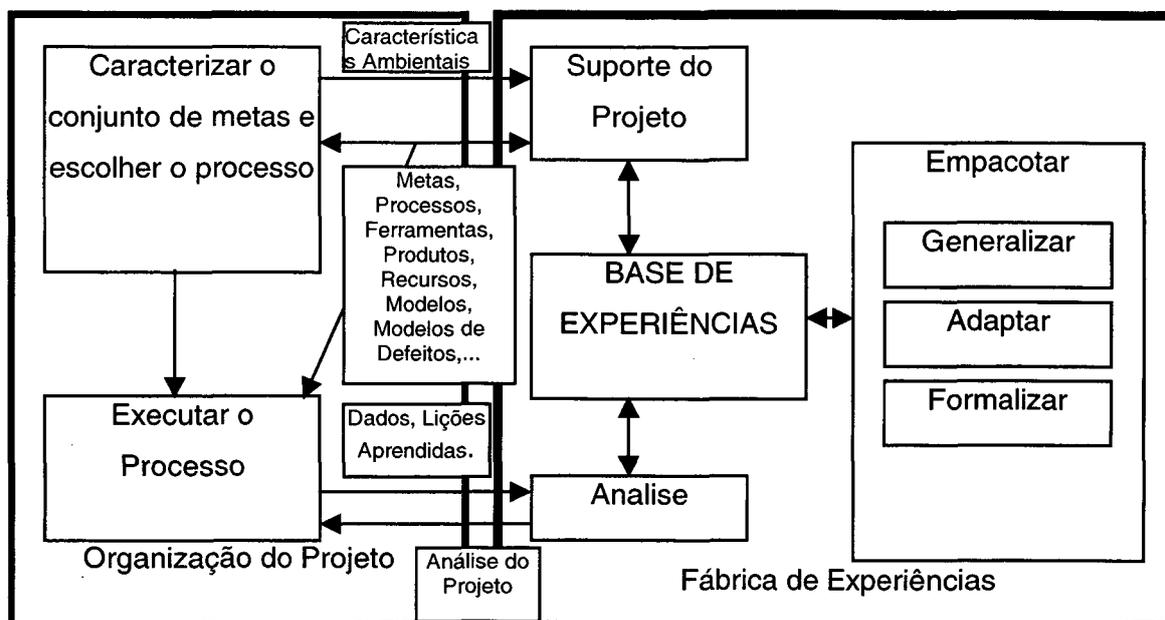


FIGURA 3. A FÁBRICA DE EXPERIÊNCIAS.

A organização de desenvolvimento, cuja meta é produzir e manter software provê a organização de análise com características do ambiente e do projeto, dados do desenvolvimento, informações sobre o uso de recursos, registros de qualidade, e informações do processo. Ela também provê *feedback* sobre o desempenho atual dos modelos processados pela fábrica de experiência e utilizados pelo projeto.

A organização de análise, pelo processamento das informações recebidas da organização de desenvolvimento, irá retornar o *feedback* diretamente para cada projeto, juntamente com metas e modelos ajustados de projetos similares.

Ela também produz e provê linhas bases, ferramentas, lições aprendidas, e dados parametrizados de alguma forma para ser adaptada às características específicas de um projeto.

A organização de suporte sustenta e facilita a interação entre desenvolvedores e analistas, pelo armazenamento e manutenção da informação, tornando-a eficientemente recuperável, controlando e monitorando o seu acesso.

A análise e interpretação dos dados coletados são baseadas nas metas. Nós podemos utilizar estes dados para: Caracterizar e entender, Avaliar e analisar, Predizer e controlar, Motivar e melhorar.

A aprendizagem sistemática requer o suporte por experiências armazenadas, generalização off-line e ajuste das experiências, e formalização da experiência. O Empacotamento de experiências úteis requer uma variedade de modelos e notações formais que são ajustáveis, expansíveis, entendíveis e acessíveis.

Uma base de experiência efetiva contém um conjunto acessível, integrado, sintetizado e empacotado de modelos de experiências que capturam experiências passadas. A reutilização sistemática requer o suporte para a utilização da experiência existente, e a necessária generalização ou ajuste da experiência candidata.

Esta combinação de ingredientes requer uma estrutura organizacional que a suporte. Isto inclui: um modelo de evolução de software que suporte reutilização, um conjunto de processos para aprendizagem, empacotamento e armazenamento de experiências, e a integração destas duas funções. A fábrica de experiências é uma unidade organizacional que realiza esta integração.

Isto cria a necessidade por organizações separadas, ao menos do ponto de vista lógico. A Organização do Projeto para o desenvolvimento do produto e a Fábrica de Experiências para o empacotamento de experiências. Ambas as organizações possuem diferentes prioridades e propósitos, usam diferentes modelos de processos, e tem diferentes necessidades de experiência. A tentativa de unir em uma mesma organização é destiná-la ao fracasso [BC91].

### 3. RACIOCÍNIO BASEADO EM CASOS

O *raciocínio baseado em casos* (RBC) [AP94, Kol93] é um paradigma para a resolução de problemas que em muitos aspectos é fundamentalmente diferente de outras atividades da inteligência artificial. Ao invés de criar associações sobre relacionamentos generalizados entre descrições de problemas e conclusões, o RBC tem capacidade para utilizar o conhecimento específico de experiências antigas e situações de problemas concretos (casos).

Um novo problema é resolvido pela busca de um caso similar no passado e sua reutilização no novo problema. Uma segunda diferença importante é que RBC é também uma abordagem para a aprendizagem sustentada e incremental, uma vez que uma nova experiência é retida a cada vez que um problema é resolvido, tornando-se imediatamente disponível para problemas futuros [AP94].

Deve-se notar que caso usualmente denota uma situação de problema. Uma situação anteriormente experimentada, anteriormente capturada e aprendida de tal forma que possa ser reutilizada na resolução de futuros problemas. Esta situação é então referenciada como um caso passado, caso armazenado ou caso retido.

Um caso típico é usualmente assumido por conter um certo grau de riqueza de informações contidas em si e uma certa complexidade com respeito a sua organização interna.

Este caso possui também outras características como: a possibilidade de ser modificado ou adaptado, e a solução recuperada podem ser aplicadas em diferentes contextos de solução de problemas [AP94]. Correspondentemente, um novo caso ou caso não resolvido é a descrição de um novo problema a ser resolvido.

Raciocínio pelo reuso de casos passados é uma forma poderosa freqüentemente utilizada por seres humanos para resolver problemas. Pessoas usam casos passados como modelos quando estão aprendendo a resolver problemas, particularmente em aprendizados recentes.

### 3.1 Fundamentos do Raciocínio Baseado em Casos

As tarefas centrais de raciocínio baseado em casos têm em comum a identificação de uma situação de problema atual, a procura por um caso no passado que seja similar ao novo caso, a utilização deste caso para sugerir uma solução para o problema corrente, a avaliação da solução proposta e a atualização do sistema pela aprendizagem desta nova experiência [AP94].

### 3.1.1 Representação de Casos

Um caso é uma peça contextualizada do conhecimento representando uma experiência. Ele contém lições passadas, isto é o conteúdo do caso e o contexto no qual a lição pode ser usada. Um caso pode ser o relato de um evento, uma estória, ou algum registro típico.

O problema descreve o estado do mundo quando o caso ocorreu, e a solução demonstra a solução derivada para um determinado problema. Em um caso é possível armazenar muitos tipos de dados no qual espera-se que possam ser armazenados em uma base de dados convencional, como nomes, identificação de produtos, valores como temperaturas e notas de texto.

### 3.1.2 O Modelo de Memória Dinâmica

O primeiro sistema que pode ser referenciado como um sistema de raciocínio baseado em casos foi o CYRUS, baseado no modelo de memória dinâmica de Schank [Sch82].

A memória do caso neste modelo é uma estrutura hierárquica a qual é chamado pacote de organização de memória episódico (E-MOP's) também referenciado como episódios generalizados (GE).

A idéia básica é organizar casos específicos os quais compartilham propriedades similares sob uma estrutura mais geral (um episódio generalizado).

Um episódio generalizado contém três diferentes tipos de objetos:

- Normas: São qualidades em comum a todos os casos sob o GE.
- Casos: Um caso pode ser o relato de um evento, uma estória, ou algum registro típico.
- Índices: São qualidades as quais discriminam os casos uns dos outros. Um índice pode apontar para um episódio generalizado mais específico ou diretamente para um caso. Um índice é composto por dois termos: O nome do índice, e o valor do índice.

A memória de casos é inteiramente uma rede discriminatória onde um nó é um episódio generalizado, o nome de um índice, o valor de um índice ou um caso.

### 3.1.3 O modelo Categoria & Exemplar

O sistema PROTOS [PB86], construído por Ray Bareiss e Bruce Porter, propõe uma forma alternativa para organizar casos em uma memória de casos. Casos são também referenciados como exemplares.

A base psicológica e filosófica para este método é a visão de que no mundo real, conceitos naturais podem ser definidos extensionalmente. Além disso, diferentes características são assinaladas com importâncias diferentes na descrição de um caso associado a uma categoria.

A memória do caso é embutida em uma estrutura de rede de categorias, relações semânticas, casos e ponteiros de índices. Cada caso é associado a uma categoria, sendo que um índice pode apontar para um caso ou uma categoria.

Os índices são de três tipos:

- ligações para características apontando de descritores do problema (características) para casos ou categorias (lembranças),
- ligações de casos apontando de categorias para os casos associados (ligação de exemplares), e
- ligações de diferenças apontando de casos para os casos vizinhos que apenas diferem em um ou um número pequeno de características. Uma característica é, geralmente, descrita por um nome e um valor.

Nesta organização de memória, as categorias são interligadas em uma rede semântica, a qual também contém as características e estados intermediários encaminhados por outros termos.

Esta rede representa uma formação do conhecimento geral do domínio o que habilita o suporte explanatório de algumas tarefas de RBC.

### 3.1.4 Indexação

Muitos sistemas de bancos de dados utilizam índices para acelerar a recuperação de dados. Um índice é uma estrutura computacional de dados que pode ser colocada na memória e rapidamente pesquisada. Isto significa que não é necessário percorrer cada registro armazenado no disco, o que pode ser lento.

O RBC também utiliza índices para acelerar a recuperação. Informações em um caso podem ser de dois tipos: Informações indexadas que serão utilizadas para recuperação e informações não indexadas que provêm informação contextual de valores para o usuário, mas não são usadas diretamente na recuperação.

Índices devem ser preditivos, endereçar os propósitos para os quais o caso será usado, ser abstrato para permitir ampliação futura da base de casos, e ser concreto para ser reconhecido no futuro [Kol93].

Existem inúmeros métodos de indexação automática, tais como:

- indexação por características e dimensões que tendem a ser preditivas por todo o domínio,
- Indexação baseada na diferença, a qual seleciona índices que diferenciam um caso de outros,
- métodos de generalização baseados em similaridade e explanação, que produzem um apropriado conjunto de índices para casos abstratos a partir de casos que compartilham um conjunto comum de características, ou
- métodos indutivos de aprendizagem, que identificam características preditivas que são então usadas como índices.

Para aplicações práticas, índices podem ser escolhidos automaticamente, manualmente, ou pelas duas técnicas [Wat97].

### 3.1.5 Armazenamento

O armazenamento de casos é um aspecto importante no desenho da eficiente de sistemas RBC, no qual é refletida a visão conceitual do que é representado no caso e dos

índices que caracterizam o caso.

Um sistema baseado em casos pode ser organizado em uma estrutura gerenciável que suporta a busca eficiente e métodos de recuperação. Um equilíbrio é encontrado entre os métodos de armazenamento que preservam a riqueza dos casos e seus índices e métodos que simplificam o acesso e a recuperação dos casos relevantes.

Estes métodos são usualmente referidos como modelos de memória de casos. Os dois mais influentes modelos de memória de casos acadêmicos são o modelo de Memória Dinâmica de Schank e Koloder, e o modelo de Categoria e Exemplar de Porter e Bareiss.

Estas técnicas são ainda largamente utilizadas pela comunidade da ciência cognitiva, mas nenhuma das ferramentas comerciais de RBC disponíveis utiliza estas técnicas.

### 3.1.6 Recuperação

A recuperação dos casos é dependente do método de representação utilizado. Em geral duas técnicas são usadas por aplicações de RBC comerciais:

#### **3.1.6.1 Nearest-Neighbor Retrieval**

Conceitualmente, é uma técnica muito simples. O que é preciso é calcular a distância relativa entre um caso alvo e os outros casos. Aquele que obtiver o menor valor é o caso vizinho mais próximo.

Esta solução utiliza-se de alguma medida de similaridade, que pode utilizar-se de pesos diferentes para cada atributo, onde a soma da similaridade de todos os atributos é calculada. Algoritmos similares a estes são usados em muitas ferramentas de RBC para realizar esta recuperação.

#### **3.1.6.2 Inductive Retrieval**

Esta técnica utilizada por muitas ferramentas RBC envolve um processo chamado indução. Indução é uma técnica desenvolvida por pesquisadores de aprendizado de

máquina para extrair regras ou construir árvores de decisão a partir de dados passados.

Em sistemas RBC, o embasamento em casos é analisado por um algoritmo de indução para produzir uma árvore de decisão que classifica ou indexa os casos.

O algoritmo de indução mais largamente utilizado é chamado de ID3 [Qui86]. Este algoritmo constrói uma árvore de decisão a partir de uma base de casos. Utiliza-se de uma heurística chamada de ganho de informação para procurar o atributo mais promissor, a partir do qual a árvore será dividida ao meio.

Como se pode ver, estas duas técnicas são amplamente usadas em ferramentas de RBC. Para decidir qual delas é melhor pode requerer experimentação e certamente requer experiência.

### 3.1.7 Adaptação

Quando um caso é recuperado, o sistema RBC estará atento para reutilizar a solução sugerida pelo caso recuperado. Em muitas circunstâncias a solução pode ser suficiente.

Porém, em outras circunstâncias a solução pode necessitar de uma adaptação da solução recuperada para as necessidades do caso atual. Adaptações procuram por diferenças proeminentes entre o caso recuperado e o caso corrente, e então aplicar formulas e regras baseadas no conhecimento do domínio, que tomam aquelas diferenças no relatório sugerindo a solução final.

Em geral existem dois tipos de adaptação em RBC:

- adaptações estruturais que aplicam regras e formulas de adaptação diretamente na solução armazenada nos casos, ou
- adaptações derivativas, a seqüência de planejamento que construiu a solução original é armazenada como um atributo adicional ao caso, e é utilizada para produzir uma nova solução para o problema corrente.

A adaptação só pode ser utilizada para domínios que estão bem entendidos.

### 3.1.8 Aprendizado

Este é o processo de incorporação que é útil para reter a partir do novo episódio de um problema resolvido na base de conhecimento existente.

A aprendizagem com sucesso ou não da solução proposta é disparada pela saída de uma avaliação e possível reparo. Isto envolve a seleção de qual informação do caso e em que formato é retido, como indexar o caso para posterior recuperação para problemas similares, e como integrar o novo caso na estrutura de memória.

A etapa de avaliação pode ser programada para a execução automática ou com participação do usuário. A avaliação tem por objetivo avaliar a qualidade da solução adaptada ao problema de entrada para definir se esta tem condições de ser adicionada a memória.

## **ANÁLISE COMPARATIVA DAS FERRAMENTAS EXISTENTES**

Neste capítulo apresentaremos uma breve descrição das ferramentas existentes que suportam o processo de planejamento baseado em GQM, apresentando suas principais características.

Todas as ferramentas aqui apresentadas encontram-se ainda na fase de protótipo, sendo, portanto, sujeitas a modificações no futuro, uma vez que passarão ainda por uma fase de validação.

Esta avaliação tem o propósito de verificar a adequação das ferramentas existentes ao trabalho dos engenheiros de qualidade na elaboração de planos de mensuração baseados no paradigma GQM, bem como de verificar se as ferramentas possibilitam além da automatização do trabalho manual, artifícios que possibilitem a estes profissionais o incremento de qualidade na elaboração de seus planos de mensuração.

### **1. GQMaspect**

A ferramenta GQMaspect [BMEK 1996] (GQM abstraction sheet and GQM plan editing and construction tool), é uma ferramenta que suporta a fase de planejamento de programas de mensuração baseados em GQM, desenvolvido no grupo de engenharia de software (AGSE) na Universidade de Kaiserslautern, Alemanha no projeto PERFECT [Per96] da Uniao European ESPRIT .

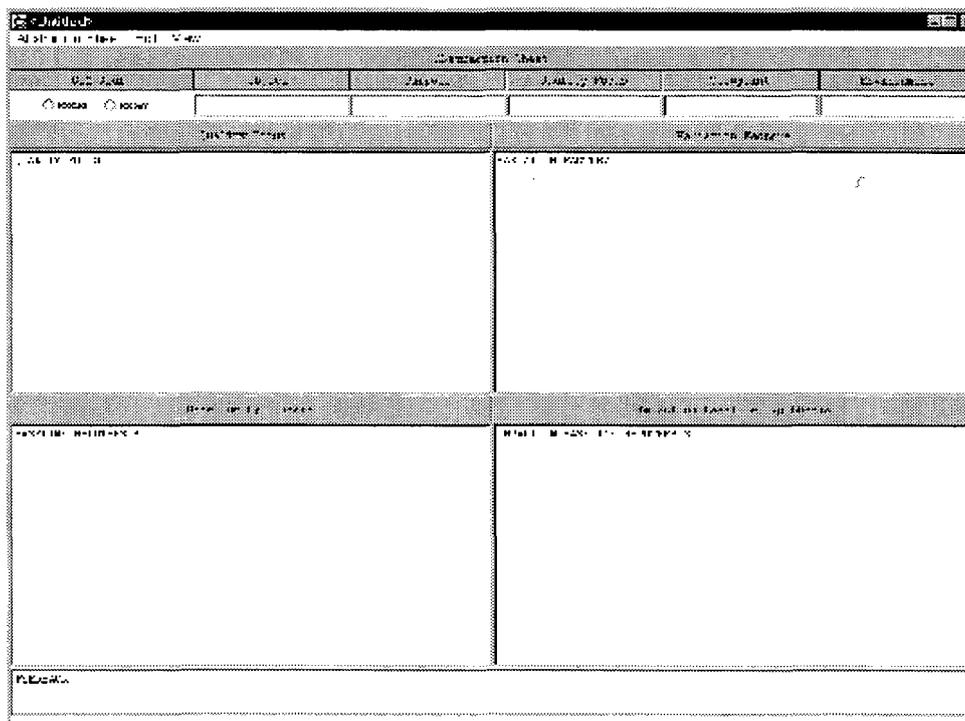


FIGURA 4. EDIÇÃO DO ABSTRACTION SHEET NO GQMAccess

Suas funcionalidades incluem:

- A definição de metas GQM.
- Construção e edição dos abstraction sheets GQM e planos GQM.
- Geração do plano GQM a partir dos Abstraction Sheets GQM e vice-versa.

O GQMaspect é um protótipo que implementa suporte à fase de planejamento de programas de mensuração baseados em GQM. É implementado na linguagem de programação Java, o que torna o sistema disponível para uma grande variedade de plataformas, tais como Sun Solaris®, Macintosh OS®, Windows 95® e Windows NT®.

A ferramenta suporta a fase de “Desenvolvimento do plano GQM” (incluindo as fases: aquisição do conhecimento, desenvolvimento das perguntas e desenvolvimento das medidas), a qual é um processo interativo. As metas GQM são definidas quando da entrada desta fase.

Primeiramente os especialistas GQM executam entrevistas com as pessoas especificadas no aspecto “ponto de vista”, da meta GQM. Os resultados destas entrevistas podem ser estruturados utilizando os abstraction sheets, um para cada entrevista.

Este modelo de Abstraction Sheet e características para sua edição estão disponíveis no GQMAspect. O próximo passo é juntar o conjunto de abstraction sheets em uma abstraction sheet sumarizada.

Quando a abstraction sheet sumarizada está completa e consistente, um plano GQM pode ser gerado a partir da abstraction sheet sumarizada utilizando o GQMAspect.

Entretanto, outras informações que não foram incluídas na abstraction sheet GQM, mas que são necessárias para a realização da mensuração baseada em GQM e, serão então inseridas no plano GQM.

Durante a construção do plano GQM, a necessidade por informação adicional pode ocorrer, o que pode tornar necessário à realização de novas entrevistas, a partir de um plano GQM já alterado.

Para tanto, o GQMAspect oferece a geração reversa das abstraction sheets GQM, a partir do plano GQM já alterado. Os abstraction sheets podem então ser atualizada com a informação adicional e o plano GQM pode ser então gerado a partir destes.



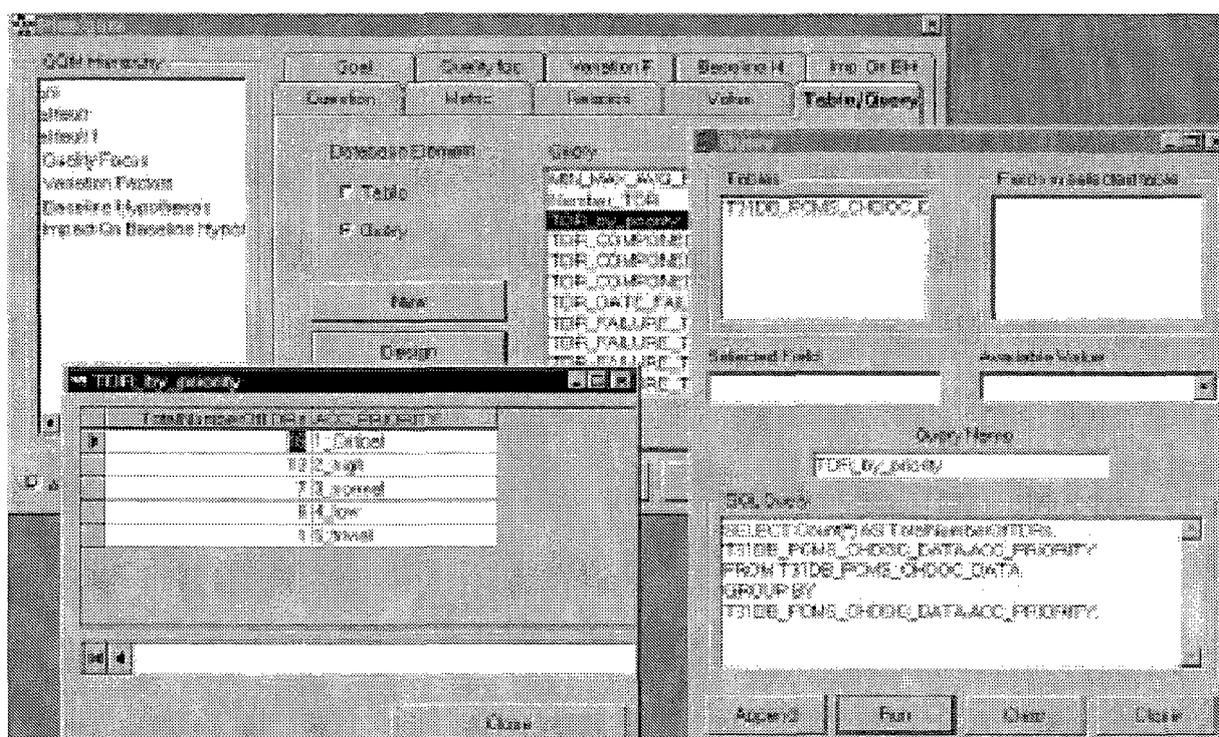
FIGURA 5. O PAPEL DO GQMAspect NO PROCESSO GQM

## 2. GQM-Tool

A primeira versão da ferramenta GQM-Tool foi desenvolvida durante o projeto CEMP [CEM96] do programa ESSI/ESPRIT da União Européia. De fato, o projeto foi iniciado sem suporte específico baseado em computador para o processo GQM, mas logo se tornou claro que a execução manual do processo GQM, e suas diversas tarefas seriam mais facilmente executadas por uma ferramenta (ver figura 6).

Algumas tarefas são suportadas como manter a definição e documentação de metas GQM, verificando a integridade referencial entre os itens do plano GQM e manter o relacionamento entre os itens e os dados coletados.

A ferramenta GQM roda sobre o ambiente Microsoft Windows. Foi escrito em MS-Visual Basic, e a base de dados de mensuração é implementada no banco de dados



relacional MS-Access.

FIGURA 6. A FERRAMENTA GQM-TOOL.

As principais características da ferramenta, e a motivação original para seu desenvolvimento, tratam com o manuseio das definições dos planos GQM:

- Edição, checagem de consistência e armazenamento dos planos GQM.
- Suporte às mudanças.
- Manutenção do relacionamento entre os itens.

Por instância, as metas de qualidade e as reutilizações definidas no projeto CEMP, ou a qualidade e metas de esforços definidos no projeto MIDAS mostram-se aplicáveis (com pequenas modificações) para diversos projetos de desenvolvimento na mesma organização.

Em geral, é muito parecido que planos GQM subseqüentes irão parecer-se com os passados compartilhando objetivos similares em alto nível. Entretanto, a habilidade de reutilizar os planos GQM existentes permitem uma significativa redução de custos no desenvolvimento e utilização de planos GQM.

Reutilizar metas significa reutilizar as definições das metas, abstraction sheets, questões, e métricas, os relacionamentos entre eles, e as ligações com os dados. Em alguns casos, mesmo a existência de dados pode ser reutilizada.

Em geral uma definição de medida corresponde a um bem definido conjunto de dados em uma base de dados de métricas. Para fazer a interpretação e a análise da coleta de dados é necessário providenciar significados para estabelecer, manter e navegar as ligações entre os dados e o plano GQM.

A existência dessas ligações faz com dos itens GQM mais compreensíveis, modificáveis e reutilizáveis.

### 3. ES-TAME

O sistema ES-TAME [Oiv90, OB92] é um protótipo de um sistema especialista para suportar o processo de desenho de sistemas de tempo real.

Um modelo sofisticado e altamente estruturado do paradigma GQM é construído dentro do ES-TAME. O usuário pode construir um plano de mensuração na forma de um conjunto de metas, questões, e métricas pela utilização de modelos para escrever metas e então selecionando um conjunto pré-definido de questões e medidas ou escrevendo novas

questões e métricas.

Raciocinando na parte do sistema especialista, é utilizado primariamente quando da construção dos planos de mensuração que são expressos como GQM's. Um gerador baseado em regras GQM utiliza o encaminhamento encadeado para guiar elementos do plano de mensuração GQM em construção.

O sistema é inteligente o suficiente para perguntar ao usuário por dados e para procurar nos modelos de qualidade para obter estes dados quando possível.

O GQM não é apenas um guia para a representação de modelos de qualidade no ES-TAME, mas o GQM é também utilizado para responder as questões e interpretar os dados coletados com os produtos de trabalho.

Quando o usuário desejar avaliar o plano de mensuração, o sistema pode automaticamente suprir o usuário com dados para aquelas métricas cujos dados existem. O sistema irá, assim, assistir o usuário na formulação de respostas para as questões com base nos valores medidos e interpretando os resultados sobre as metas.

#### 4. GQM-DIVA

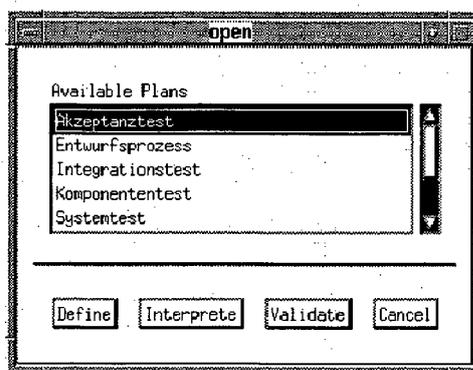
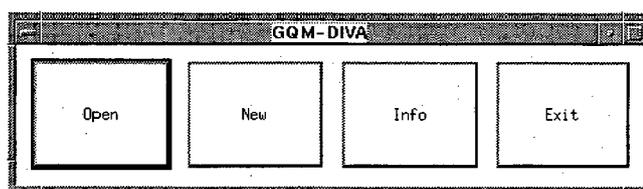
O GQM-DIVA [Mar95, Dif93] é uma ferramenta (ver figura 7) que suporta a definição de planos GQM através de uma interface gráfica de usuário, a validação dos planos GQM em termos de falta ou inconsistência de elementos, e a interpretação dos dados coletados através de procedimentos definidos pelo usuário.

FIGURA 7. INTERFACE DO GQM-DIVA

A ferramenta GQM-DIVA suporta as fases de definição de Metas, modelos, questões e métricas que podem ser sucessivamente refinadas, e foi desenvolvida para o sistema operacional SunOS Vrs. 4.1, com X-Window System release 5 e OSF/Motif Vrs. 1.2.1.

## 5. GQM-PLAN

O objetivo final do GQM-PLAN [AK99] é dar suporte à elaboração do Plano GQM,



através da criação de uma base de dados que contenha os objetivos da avaliação, o foco de qualidade, os fatores de variação, as hipóteses básicas, os impactos sobre as hipóteses básicas, e as questões e métricas relacionadas ao programa de avaliação que está sendo desenvolvido.

A ferramenta GQM-PLAN destina-se tanto aos profissionais responsáveis pela realização do programa de qualidade quanto aos membros da empresa que estão envolvidos no programa em questão. A arquitetura básica da ferramenta compreende os módulos de entrada de dados, consistência e emissão de relatórios, e foi implementada para funcionar em computadores pessoais com ambiente operacional windows.

## 6. Comparação entre as ferramentas apresentadas

Apresentamos a seguir uma breve comparação entre as ferramentas disponíveis e sua adequação ao processo de planejamento de mensuração baseado em GQM, bem como facilidades disponíveis, tais como a possibilidade de reutilização de conhecimento, e automação de tarefas manuais.

Característica / Ferramenta	GQM-Aspect	GQM-Tool	ES-TAME	GQM-Diva	GQM-Plan
<b>Suporta todos os passos do planejamento</b>					
Definição da Meta	☺	☺	☺	☺	☺
Aquisição do Conhecimento	☺	☺	☹	☹	☺
Definição das Perguntas	☺	☺	☺	☺	☺
Definição dos Modelos	☹	☹	☹	☹	☹
Definição das Medidas	☺	☺	☺	☺	☺
Definição dos Procedimentos de Mensuração	☹	☹	☹	☹	☹
Definição dos Instrumentos de Mensuração	☹	☹	☹	☹	☹
<b>Suporta recuperação de experiências</b>					
Produtos GQM	☹	☹	☺	☹	☹
Lições aprendidas	☹	☹	☹	☹	☹
Recuperação baseada em similaridade	☹	☹	☹	☹	☹

FIGURA 8. COMPARAÇÃO ENTRE AS FERRAMENTAS.

Como pode ser observada na figura 8, a grande maioria dos sistemas apenas auxiliam na edição dos planos de mensuração baseados em GQM concentrando-se apenas no auxílio da definição das metas de mensuração e o seu refinamento em perguntas e posteriormente em medidas sem, entretanto, auxiliarem na composição do plano efetivo de mensuração.

A reutilização dos produtos desenvolvidos nos planos é pouca ou nenhuma, na grande parte das ferramentas, não levando em consideração o conhecimento existente nos planos de mensuração criados com o auxílio das ferramentas.

O uso do conhecimento de lições aprendidas através do registro de problemas e soluções executadas não é de forma alguma suportado pelas ferramentas apresentadas, deixando a cargo dos especialistas a análise e solução dos problemas que por ventura

venham a ocorrer durante o processo de planejamento.

Devido a pouca ou nenhuma reutilização de produtos ou conhecimento existente nos planejamentos efetuados, nenhuma das ferramentas implementa um mecanismo de recuperação de conhecimento baseado em similaridade, o que poderia ser de grande valia aos engenheiros de garantia de qualidade na execução do processo de planejamento de programas de mensuração.

## ABORDAGEM BASEADA EM CASOS PARA REUTILIZAÇÃO DE EXPERIÊNCIAS DE MENSURAÇÃO

Neste capítulo apresentaremos uma visão geral sobre a abordagem baseada em raciocínio baseado em casos para operacionalização de uma fábrica de experiências com o aprendizado, aquisição sistemática armazenamento e comunicação por toda a organização de experiências de mensuração em ambientes industriais [Gre99, GAB99, GB98,GAB98].

Nesta abordagem, experiências de mensuração são acumuladas na *base de experiência GQM* (GQM-EB) [GAB99], refletindo o conhecimento de mensuração específico da organização, o qual foi adquirido durante os programas de mensuração passados.

Estas experiências podem ser solicitadas durante o planejamento de um novo programa de mensuração para guiar, suportar e melhorar o presente processo de planejamento.

Experiências relevantes são definidas baseadas nas características do contexto, como por exemplo, departamentos, problemas e metas de projetos de software.

Casos similares são recuperados da base de experiências para o usuário através de um sistema de navegação, o qual permitem uma exploração interativa dos casos candidatos.

A partir do conjunto de casos candidatos à reutilização, o mais apropriado é selecionado pelo usuário, utilizado como uma base inicial para o processo de planejamento atual e,se necessário, adaptado para adequar-se às necessidades específicas do projeto de

software atual.

Desde que a GQM-EB é utilizada como meio de comunicação para compartilhar experiências por toda a organização instantaneamente, a recuperação e a reutilização de experiências são enfatizadas ao invés de adaptação automática.

Novas experiências são adquiridas e integradas na GQM-EB, cada vez que um programa de mensuração é planejado como parte inerente do processo de reutilização.

### 1. Representação de Experiências de Mensuração

Para o suporte baseado nas experiências do planejamento de programas GQM na prática, experiências de mensuração são representadas na GQM-EB. Para prover um suporte efetivo, a GQM-EB deve capturar vários tipos de experiências de mensuração.

Isto inclui conhecimento de produtos específicos desenvolvidos durante o planejamento de programas de mensuração, lições aprendidas através do processo de planejamento, assim como terminologias e conceitos de mensuração utilizados em uma organização específica e suas interdependências.

Para ser efetivo, experiências GQM serão adaptadas à um ambiente organizacional específico. No contexto da mensuração de software, cada projeto de software contribui para o conhecimento organizacional pela captura de experiências ajustadas a este projeto.

As experiências específicas do projeto são acompanhadas com a caracterização do contexto de um projeto em particular, o qual permite a recuperação de experiências de projetos de software similares.

O conhecimento é primariamente representado na forma de casos concretos, os quais são descrições específicas do contexto de experiências ganhas durante o planejamento de programas de mensuração passados.

Para estender o escopo da aplicação, padrões genéricos podem ser representados integrando experiências específicas de projetos em comuns. A reutilização é

suportada pelo conhecimento geral do domínio na terminologia da mensuração e conceitos para suporte adicional.

## 2. Casos de Experiências

Dois tipos principais de casos de experiências de mensuração são representados:

- Casos de experiências de produto GQM (GQM-PEC): Estes casos incluem todos os produtos GQM desenvolvidos durante o planejamento de um programa de mensuração baseado em GQM. Eles são reutilizados em projetos de software similares como base para o desenvolvimento dos respectivos produtos. Isto resulta na redução do esforço de planejamento e melhoria da qualidade dos produtos GQM através da confiabilidade, completude e consistência.
- Casos de experiências de problemas e soluções GQM (GQM-PSEC): Estes casos capturam explicitamente as estratégias de problemas e soluções que foram adotadas em programas de mensuração passados, seu contexto de uso, e informação sobre o seu grau de sucesso. GQM-PSECS podem alertar sobre falhas potenciais e suportar a busca por soluções no contexto da aplicação.

### 2.1. Casos de experiências de produtos GQM (GQM-PEC)

Os produtos GQM das GQM-PECS são representados por um conjunto de atributos relevantes e interdependências baseadas em modelos de domínio de produtos GQM.

Para recuperar os casos de experiências de projetos de software similares, uma caracterização do ambiente a partir do qual o caso será obtido é essencial.

Os contextos das experiências são descritos por um conjunto mínimo de características as quais permitem identificar casos similares e discriminantes através do processo, objetivo e ambientes relacionado para reutilização.

Para avaliar o potencial de reutilização dos casos e prever a reutilização de experiências inadequadas, casos são estendidos por informações básicas. Informações

sobre a reutilização de casos em planejamentos de mensuração como precondições para reutilização, adaptações necessárias, o custo de reutilização e a frequência de reutilização são explicitamente capturados.

## 2.2. Casos de experiências de problemas e soluções GQM (GQM-PSEC)

Para capturar explicitamente estratégias de problemas e soluções que ocorreram durante o planejamento de programas de mensuração passados, GQM-PECs são modeladas e representadas como casos sobre problemas encontrados enquanto se desenvolvia ou utilizava produtos GQM.

Em geral, um GQM-PSEC inclui uma descrição do contexto, se o problema ocorreu durante o planejamento de um programa GQM, uma lista de causas do problema, a solução aplicada, o resultado obtido da solução e informações básicas.

## 2.3 Conhecimento geral do domínio

Para suportar a reutilização apropriada de experiências e facilitar a representação e a aquisição consistente de novas experiências durante os projetos de software, conhecimentos gerais do domínio através das experiências GQM são representados na GQM-EB.

### 2.3.1 Modelos de produtos GQM

Produtos GQM são explicitamente modelados, na forma de objetos, em classes hierárquicas.

Cada classe é estruturada por um conjunto de atributos atômicos ou relacionamentos com outros objetos.

Atributos são definidos por um identificador, descrição, cardinalidade, seu tipo ou relacionamento e um valor padrão.

### 2.3.2 Definição de Tipos

Definições de tipos modelam qualidades de entidades como experiência do desenvolvedor, ou categoriza conceitos, como linguagens de programas.

Por exemplo, a escala de mensuração de um conceito de medida GQM é definida como um símbolo ordenado com os valores possíveis: "nominal", "ordinal", etc, usando uma medida específica de similaridade para determinar a similaridade entre os valores definidos.

Definições de tipos são usadas para definir os tipos aos atributos de classes. Seu suporte de aquisição consistente de experiências durante os projetos facilita a avaliação da situação, e também suporta o processo de adaptação manual da recuperação de casos de experiências pela indicação explícita de alternativas.

Num contexto específico, tipos podem ser derivados de um conjunto predefinido de tipos básicos, como por exemplo: texto, numérico, lógico, taxonomias, etc.

Para cada tipo uma similaridade local é definida. Para tipos simbólicos, o significado de possíveis valores é explicitamente definido pela definição de faixas.

### 2.3.3 Glossários

Glossários definem a terminologia e os conceitos básicos relacionados à mensuração de software. Um glossário suporta o uso adequado de termos, sua consistência através da organização, e assegura que a reutilização de produtos GQM é baseada em suas suposições.

## 4. Suporte baseado em experiências para planejamento GQM

O planejamento de programas de mensuração GQM é suportado pela reutilização de programas de mensuração passados. Enquanto o responsável GQM está planejando um programa de mensuração baseado em GQM, ele pode requisitar suporte baseado em

experiências.

Durante cada tarefa do processo de planejamento, casos apropriados descritos em experiências passadas de projetos de software similares são recuperados da GQM-EB. Aquele que melhor supre as necessidades atuais pode ser utilizado como uma base inicial para o desenvolvimento de produtos GQM ou para suportar a performance da tarefa nos programas de mensuração atuais.

A elucidação das características do contexto é interativamente guiada pela GQM-EF, a qual infere automaticamente a informação de entrada, com base nos produtos desenvolvidos nos passos anteriores, o que faz com que o usuário necessite informar dados de entrada apenas onde não é possível inferi-los automaticamente, como é o caso da recuperação de soluções para problemas ocorridos durante o processo de planejamento.

Este modelo é baseado nas características utilizadas no passado para descrever o ambiente e envolve continuamente o usuário que pode adicionar ou remover características quando ele achar relevante ou irrelevante para a caracterização.

As entradas das características do contexto podem ser suportadas pelo conhecimento geral do domínio. Glossários podem ser usados para o uso consistente de terminologias pelo projeto e taxonomias para guiar e direcionar a definição apropriada das características.

Definições de faixas facilitam a atribuição de valores e garantem a descrição consistente através dos diversos projetos.

Para descrever a atual situação, certas características da descrição do contexto são absolutamente essenciais, outras são de menor importância. Para refletir a relevância das características da descrição do contexto atual, um fator de relevância é eleito para característica.

O fator de relevância declara a importância de um atributo em particular para a determinação da similaridade entre o contexto atual e a descrição do contexto de um caso na GQM-EB. Aqui, os fatores de relevância são classificados como essencial, importante, menos importante, e irrelevante.

De acordo com o fator de relevância, cada característica do contexto pode ser associada a um peso usado para computar a similaridade por técnicas adequadas de RBC, similar ao sistema PATDEX [Wes95].

#### 4.1 Recuperando casos de experiências relevantes

Quando as experiências de programas de mensuração passados são reutilizadas para facilitar o planejamento de um atual, casos apropriados devem ser identificados na GQM-EB.

Assim a utilidade de experiências passadas pode somente ser determinada quando se tenta reutilizá-la no projeto atual, um critério posterior de utilidade é reduzido para um critério, a priori, de similaridade entre a caracterização do contexto dos casos, assumindo que a similaridade da caracterização do contexto também implica na utilidade da solução. Isto não é uma tarefa trivial, uma vez que considera a avaliação e comparação de representações complexas do conhecimento.

Entretanto, inclui a manipulação de informações incompletas, assim como no domínio de mensuração de software certas características do contexto podem ser desconhecidas, quando iniciamos um programa de mensuração, ou casos na GQM-EB podem ter sido descritos incompletamente, por causa de certas características do contexto não terem sido reportadas no passado.

Para a recuperação de candidatos a reutilização, a GQM-EB realiza uma busca primária, comparando exatamente as características do contexto atuais marcadas como essencial e as respectivas características armazenadas na GQM-EB.

O conjunto resultante de casos é então avaliado através do grau de similaridade das características restantes marcadas como importantes ou menos importante. Características as quais não são especificadas ou marcadas como irrelevantes são descartadas do cálculo de similaridade.

A determinação da similaridade pode ser efetuada aplicando medidas de

similaridade [Ric95]. Medidas de similaridade determinam para cada caso um valor de similaridade numérico e induz uma ordem parcial entre os casos na base de casos.

Medidas de similaridade globais são baseadas no número de atributos com os valores correspondentes. Para prevenir que vários casos possam ter valores de similaridade baixos, por apenas alguns poucos componentes serem correspondentes, embora eles possam ser completamente similares, medidas de similaridade global devem ser aplicadas.

Medidas de similaridade local também consideram a similaridade dos valores dos atributos. Refletindo os diferentes graus de importância de características individuais para a determinação dos valores de similaridade, os fatores de relevância dados serão integrados na medida de similaridade.

Para prevenir a sobrecarga do usuário com informações inúteis, casos com pequeno grau de similaridade são excluídos utilizando-se um limite para que os casos sejam considerados suficientemente similares.

Além da recuperação de casos de projetos específicos, também padrões genéricos podem ser recuperados. Padrões genéricos diferenciam-se de casos de projetos específicos pelo nível de abstração dos valores das características do contexto.

Daí, se o usuário provê valores mais abstratos como entrada para a recuperação, mais padrões genéricos serão recuperados do que valores específicos, resultando em um conjunto de candidatos a reutilização de projetos específicos.

Os diferentes níveis de abstração dos valores são apresentados nas definições de faixa por taxonomias, as quais são utilizadas para determinar a similaridade.

#### 4.1.1 Manutenção das metas de recuperação.

Com base em cenários de reutilização, metas de recuperação [GAB99] são determinadas. Metas de recuperação (ver figura 10) demonstram explicitamente o objeto a ser reutilizado, o propósito da reutilização, a tarefa relacionada, o ponto de vista específico e o ambiente em particular.

<b>Modelo de Meta</b>	<b>Exemplo</b>
Recuperar <objeto>	lições
para o <propósito>	guiar a solução de problemas
a respeito do <processo>	mensuração de software
a partir do <ponto de vista>	peçoal de garantia de qualidade
no contexto de <ambiente>	Companhia XYZ

FIGURA10. MODELO DE META DE RECUPERAÇÃO.

Com base nas metas de recuperação, fatores de reutilização são determinados através de entrevistas com especialistas do domínio. Isto inclui a especificação de conceitos relacionados, índices relevantes, e suas interdependências.

Objetos definidos na meta de recuperação são explicitamente descritos em um glossário de conceitos [TG99], descrevendo seu propósito e usuários.

Atributos relevantes de objetos utilizados como índices no processo de recuperação são identificados, com base nas seguintes considerações:

- Quais são os atributos/características para descrever o objeto?
- Quais informações são necessárias para a execução da tarefa no que diz respeito ao propósito específico?
- Quais informações são necessárias para a identificação e seleção dos objetos adequados?
- Quais atributos estão habilitados para discriminar objetos?
- Quais qualidades mínimas os objetos precisam ter?

Para cada um dos índices identificados, seu tipo, faixa e cardinalidade são explicitamente definidos [TG99] para possibilitar uma representação consistente do conhecimento e para avaliar a situação de suporte.

Para a recuperação de experiências de engenharia de software, uma medida de similaridade genérica é desenvolvida, a qual é ajustada para cada meta de recuperação considerando a similaridade global e local, relevância dos índices e imperfeição do conhecimento.

#### 4.2 Suporte interativo para a reutilização de casos de experiências

Os casos candidatos associados com o grau de similaridade com a caracterização do contexto dada são apresentados para o usuário por meio de um sistema de navegação.

Isto permite a exploração interativa dos casos pelo usuário, facilitando a seleção do caso mais apropriado, suportando a adaptação do caso selecionado às necessidades atuais e, se necessário a execução de uma recuperação refinada.

Isto permite a utilização da GQM-EF sem necessidade do conhecimento extensivo do domínio e o entendimento da estrutura de representação interna. Decisões informadas são então suportadas pela captura explícita de experiências sobre a reutilização de um caso em particular no passado, como custo de adaptação, problemas ocorridos, etc.

Se o sistema falha ao propor casos, o conhecimento geral do domínio estão disponíveis para facilitar o planejamento do programa de mensuração.

#### 4.3 Aquisição do conhecimento específico do projeto

A evolução incremental baseada no retorno de aplicações industriais é essencial para o melhoramento contínuo das tecnologias de engenharia de software.

A respeito de programas de mensuração, isto pode continuamente reduzir o esforço relacionado à fase de planejamento e melhorar a consistência e a efetividade da mensuração.

Conseqüentemente, o conhecimento na GQM-EB é melhorado e atualizado a cada vez que um novo programa de mensuração é estabelecido. Cada vez que uma experiência GQM é reutilizada para suportar o planejamento de um programa de mensuração, novas experiências neste programa de mensuração são capturadas.

Para manter o esforço relacionado à aquisição do conhecimento mínimo, este

processo é intercalado com o processo de reutilização: Informações providas pelo usuário como entradas para o processo de recuperação e experiências recuperadas da GQM-EB são reutilizadas durante a aquisição para otimizar o processo.

Enquanto o usuário solicita suporte baseado em experiência, ele provê uma caracterização preliminar do contexto da situação presente para recuperar casos candidatos a reutilização apropriada da GQM-EB.

Paralelamente ao processo de recuperação, esta informação é capturada em um novo caso documentando a experiência atual. GQM-PECs reutilizados para o desenvolvimento de produtos GQM no programa de mensuração atual ou GQM-PECs reutilizados para a solução de problemas atualmente encontrados são utilizados para suplementar o novo caso.

Enquanto captura um novo caso, ele é melhorado por informações básicas, as quais são perguntadas ao usuário. A descrição do novo caso deverá ser cuidadosamente revisada pelo usuário; informações ainda faltantes serão adicionadas e derivações de casos reutilizados serão modificadas.

A aquisição manual de novas experiências é guiada através de uma estrutura de caso detalhada, a qual endereça explicitamente as dimensões relevantes a serem capturadas.

O uso do conhecimento organizacional do domínio, como glossários e definição de tipos, facilitará a descrição consistente de experiências através dos projetos de software. Se um caso da GQM-EB é reutilizado, sua informação de reuso será atualizada pela obtenção das respectivas informações do usuário.

A aquisição de informação para a atualização e melhoramento do conhecimento geral do domínio na GQM-EB serão também integradas ao processo de reutilização.

Novas entidades utilizadas durante o programa de mensuração atual, podem ser adicionadas pelo usuário aos glossários, taxonomias e definições de faixa, melhorando o conhecimento geral do domínio continuamente.

#### 4.4 Integração de novas experiências

Os casos específicos de projetos adquiridos recentemente e o conhecimento geral do domínio devem ser integrados na base de experiências existentes e disponibilizado para reutilização futura.

A integração de novas experiências representa o processo de aprendizagem. Isto inclui a seleção de informações a serem capturadas, sua representação apropriada na GQM-EB e a identificação de características relevantes do contexto para indexação.

Isto implica que os casos devem ser armazenados, as interdependências criadas ou atualizadas, e se necessário padrões genéricos de casos serão criados ou modificados. Adicionalmente, o conhecimento geral do domínio será atualizado e melhorado.

Casos específicos de projetos são armazenados na GQM-EB; Os relacionamentos entre as entidades de um caso são criados; Os casos específicos de projetos são avaliados através de sua similaridade com outros casos na GQM-EB.

Se o novo caso difere apenas em pequenos detalhes do caso reutilizado, um caso abstrato substitui os dois casos específicos de projetos criando uma generalização.

O desenvolvimento de padrões genéricos através de um engenheiro de conhecimento pode ser guiado pelas taxonomias relacionadas o que provê uma base para a derivação de abstrações de valores específicos.

A integração de novas experiências pode também requerer a modificação da estrutura do conhecimento e dos esquemas de indexação.

Uma vez que as características relevantes do contexto utilizadas como índices dependem de uma organização específica e pode mudar no decorrer do tempo, o ajuste contínuo dos esquemas de indexação baseados nas características relevantes do contexto deve ser suportado durante todo o ciclo de vida de uma GQM-EB

## 5. Recuperação de experiências de Engenharia de Software

Para possibilitar suporte baseado em experiências para tarefas de engenharia de software, o processo de recuperação inclui os seguintes passos:

### 5.1 Avaliação da situação

A situação atual é descrita pelo usuário, especificando a meta de recuperação e o conjunto de índices relacionados com a meta de recuperação. A importância de cada índice da meta de reutilização específica é caracterizada através de fatores de relevância pré-definidos assinalados a cada índice, os quais podem ser manualmente adaptados para a situação específica pelo usuário.

### 5.2 Busca exata dos índices marcados como essenciais

Num primeiro passo, os casos na base de experiências são comparados com a descrição da situação pelos índices marcados como essenciais. Pela comparação perfeita destes índices, um conjunto de candidatos potenciais de reutilização é determinado.

### 5.3 Comparação parcial dos casos

Para todos os candidatos potenciais de reutilização, um valor de similaridade é calculado pela comparação parcial dos índices, (exceto os marcados como essenciais) utilizando uma medida de similaridade específica da meta de recuperação. Casos com o valor de similaridade maior do que o valor limite dado, são considerados suficientemente similares e os casos mais similares são propostos para o usuário como candidatos para a reutilização.

#### 5.4 Seleção dos candidatos a reutilização

Baseado nos casos propostos para reutilização associados ao seu valor de similaridade, o usuário pode explorar os casos, selecionando o caso mais apropriado para reutilização e, se necessário adaptá-lo para as necessidades atuais.

##### 5.4.1 Medida de similaridade para recuperação de experiências de engenharia de software

Nesta seção, apresentaremos uma medida de similaridade genérica, para a identificação de experiências de engenharia de software similares na base de experiências que podem ser ajustadas para uma meta de reutilização específica.

A medida de similaridade é utilizada para assinalar um valor de similaridade para os casos armazenados, comparando-os parcialmente com a descrição da situação dada. Basicamente, uma medida de similaridade  $\text{sim}(x,y)$  pode ser definido por [Alt97, Ric95, Wes95]:

Um mapeamento  $\text{sim}: M \times M \rightarrow [0,1]$  em um conjunto  $M$  é a medida de similaridade se (Para todo  $xy \in M$ ):

- \*  $\text{sim}(x,y) = 1$  (reflexividade)
- \*  $\text{sim}(x,y) = \text{sim}(y,x)$  (simetria)
- \*  $\text{sim}(x,y) = 1 \leftrightarrow x=y$

Dependendo da meta de recuperação, um conjunto particular de índices é utilizado para avaliar a nova situação e compará-los aos casos armazenados na base de experiências.

Um conjunto de índices  $C$  é representado como uma lista de características  $C=\{C_1,C_2,\dots\}$ . A faixa de valores  $c_i$  da característica  $C_i$  é definida por  $W_i$ , sua respectiva definição da faixa armazenada na base de experiências. A situação atual é avaliada com base no conjunto de índices da meta de recuperação.

A descrição da situação atual é representada como uma lista de pares característica/valor  $S_i = \{(C_1, S_1), (C_2, S_2), \dots\}$  incluindo a característica  $C_i \in C$  e seus valores  $S_i \in W_i$ .

Na base de casos, um caso  $k = (E_k, e_k)$  representa uma experiência pelo par característica/valor (experiência  $E_k = \{(E_{k1}, e_{k1}), (E_{k2}, e_{k2})\}$  com as características  $E_{ki}$  e seus valores  $e_{ki} \in W_i$ , descrevendo o conhecimento ganho em projetos de software passados, o contexto do qual se originou, e suas interdependências com outros produtos, processos, e recursos e um limite  $e_k \in [0, 1]$  que definem um limite do valor global de similaridade, determinando se o caso é suficientemente similar com a situação a ser proposto ao usuário como candidatos a reutilização.

#### **5.4.1.1 Recuperação orientada a metas**

Na maioria dos sistemas RBC, os índices são definidos como uma parte fixa do caso armazenado na base. Como no domínio da engenharia de software, muitos índices dependem da meta de recuperação em particular.

Os índices são relacionados ao tipo específico da recuperação. Uma meta de recuperação específica  $g$  está associada a um conjunto particular de índices  $C_g = \{C_{g1}, C_{g2}, \dots\}$ .

Entretanto, uma medida de similaridade genérica é definida, utilizando um conjunto de índices como parâmetros, que podem ser instanciados a partir de uma meta de recuperação específica.

#### **5.4.1.2. Manipulando dados incompletos**

Como atualmente, modelos de domínio da engenharia de software não estão disponíveis na prática, precisamos tratar a informação incompleta, a respeito da entrada, assim como os casos armazenados na base de casos.

Nossa abordagem está baseada no modelo de contraste de Tversky [Tve77], que foi aplicado no sistema PATDEX [AW91]. Este modelo expressa a similaridade de dois objetos através de um contraste linear de pesos diferentes entre suas características

comuns e diferentes.

De acordo com as situações potenciais que podem ocorrer quando comparando a situação dada e o caso armazenado, os seguintes conjuntos de características são distinguidos:

\* E: Conjunto de características correspondentes. O valor da característica da situação dada corresponde com um do caso armazenado na base.

\* W: Conjunto de características contraditórias. O valor da característica da situação dada não corresponde com um do caso armazenado na base.

\* U: Conjunto de características desconhecidas. A característica está apenas contida no caso armazenado na base, mas não está representado na descrição da situação atual.

\* R: Conjunto com características redundantes: A característica está descrita na situação atual, mas não está contida no caso armazenado na base. Os casos podem ter sido descritos incompletamente no passado por várias razões, p.ex. tempo esgotado ou falta de conhecimento do domínio.

A subdivisão em diferentes conjuntos de características permite a manipulação explícita de informação desconhecida e características correspondentes / contraditórias durante o processo de recuperação pela associação com cada um destes conjuntos de características com diferentes pesos ( $\alpha, \beta, \gamma, \delta \in [0,1]$ ).

Dependendo de como escolhemos uma estratégia otimista ou pessimista, podemos atribuir pesos maiores a características correspondentes do que os contraditórios, ou vice e versa, respectivamente.

No domínio da engenharia de software, onde companhias estão atualmente iniciando a construção de bases de experiências e pequena quantidade de experiências disponíveis, nós sugerimos uma estratégia otimista com  $\alpha > \beta$ , atribuindo pesos maiores às características correspondentes do que as não correspondentes.

Características desconhecidas são consideradas menos importantes para a identificação de casos relevantes ( $\gamma > 0$ ). Características redundantes acredita-se ter um impacto na determinação de casos relevantes. Mas como os valores específicos não estão disponíveis nos respectivos casos na base de casos, eles são associados a um peso muito pequeno ( $\delta \ll \beta$ ).

Os pesos dependem do ambiente específico e podem mudar durante o ciclo de vida na base de experiências.

#### **5.4.1.3. Relevância dos índices para similaridade**

A recuperação de experiências da base de dados somente será útil se a experiência recuperada puder ser aplicada atualmente na situação dada. O problema aqui é identificar características relevantes para a recuperação dos casos apropriados, antes de tentar reutilizar esta experiência na situação dada.

Baseado no sistema PADTEX [AW91], fatores relevantes são definidos, os quais determinam a importância da característica. Entretanto, no contexto da Engenharia de Software fatores relevantes devem ser relacionados a uma meta específica de recuperação.

Então para cada índice  $Cg_i$  que pertence ao conjunto  $Cg$  para uma meta de recuperação específica, um fator de relevância  $\omega g_i \in [0,1]$  é definido. Para cada meta de recuperação, estes fatores de relevância são representados pela significação do vetor de relevância  $Rg = \{\omega g_1, \omega g_2, \dots\}$  com  $\sum \omega g_i = 1$  normalizada.

O fator de relevância representa um peso usado para o cálculo do valor de similaridade. Os fatores de relevância podem ser aprendidos durante o passar do tempo através de uma abordagem pseudoconexionista [Alt97] com base em uma abstração da aprendizagem competitiva de Hebb tomada do reconhecimento de padrões.

Definindo inicialmente todos os índices igualmente relevantes, durante o processo de aprendizagem os pesos das características que recebem um retorno positivo são fortalecidos, ou enfraquecidos no caso de retorno negativo.

Adicionalmente, o usuário pode sobrescrever manualmente os fatores de

relevância se necessário. Esta é facilitada através da classificação dos fatores nas seguintes categorias e pesos associados:

- Essenciais: Índices que combinam exatamente com o valor dado. Através da declaração dos índices essenciais o usuário define filtros. Casos nos quais os valores das características definidas como essenciais não coincidirem serão rejeitados a despeito dos valores das outras características.
- Importante / Menos importante: Índices que combinam parcialmente com a situação descrita.
- Irrelevantes: Índices são desconsiderados na recuperação. ( $\omega_i = 0.0$ ).

Sistematicamente capturando o retorno das modificações manuais dos fatores de relevância pelo usuário habilita o aprendizado contínuo e ajuste dos vetores de relevância.

#### **5.4.1.4 Similaridade local das características**

Para um exame compreensivo da similaridade entre a situação dada e os casos na Base de Experiências, a similaridade local entre os valores da característica deve ser considerada.

Então, uma medida de similaridade local específica deve ser definida para cada tipo de índice. Aqui, medidas de similaridade local genéricas  $v'(v_i, v_j) \in [0,1]$  para os tipos básicos de valores  $W(v)$  no domínio da engenharia de software [Gre98, TG99] são providos, os quais ainda devem ser refinados para um ambiente específico na prática.

Deixar  $V_i$  e  $V_j$  são dois valores da característica:  $V_i$  representa o valor da situação dada e  $V_j$  representa o valor respectivo do caso armazenado na base de casos. Além disso, as características são representadas pelo valor atômico.

Assim, os valores das características são interpretados como um conjunto com um número mínimo e máximo de elementos definidos por sua cardinalidade. O valor de similaridade local é computado com base na soma normalizada das melhores combinações

para cada elemento do conjunto de valores da situação avaliada com o caso [TG98]:

se  $\text{card}(v_i) > 0$  e  $\text{card}(v_j) > 0$  então

$$v'(v_i, v_j) = \frac{\sum_{e_2 \in v_j} (\max(v(e_1, e_2) \mid e_1 \in v_i))}{\text{card}(v_j)}$$

senão

$$v'(v_i, v_j) = 0$$

Nas seguintes seções, medidas de similaridade local  $v'(v_i, v_j)$  para o cálculo do valor de similaridade entre dois valores atômicos de características são descritas para os tipos básicos [GAB99].

- **Números:** A similaridade entre os valores numéricos pode ser descrita como uma unção assintótica [Ber+97], onde a diferença entre dois valores causa um decréscimo assintótico da similaridade. Para tolerar diferenças entre os valores até a distância definida  $x_1$  a seguinte função de similaridade pode ser utilizada:  $v(v_i, v_j) = \exp(-\alpha |v_i - v_j|)^3$  com  $\alpha \in \mathfrak{R}$ .
- **Intervalos:** A similaridade de dois valores descritos por intervalos pode ser calculado a respeito de suas bordas [TG99]: Sendo  $v_i = [l(v_i), h(v_i)]$  e  $v_j = [l(v_j), h(v_j)]$  intervalos:
 
$$\begin{aligned} v(v_i, v_j) &= (l(v_j) - l(v_i) + l(v_j) - h(v_j)) / 2 \text{ se } h(v_i) < l(v_j); \\ v(v_i, v_j) &= (l(v_j) - l(v_i)) / 2 \text{ se } l(v_i) < l(v_j) \leq h(v_i) \leq h(v_j); \\ v(v_i, v_j) &= 1 \text{ se } l(v_j) < l(v_i) \wedge h(v_i) \leq h(v_j); \\ v(v_i, v_j) &= (h(v_i) - h(v_j)) / 2 \text{ se } l(v_j) < l(v_i) \leq h(v_i) \leq h(v_j); \\ v(v_i, v_j) &= (l(v_i) - h(v_j) + h(v_i) - h(v_j)) / 2 \text{ se } h(v_j) < l(v_i); \\ v(v_i, v_j) &= (l(v_j) - l(v_i) + l(v_i) - h(v_j)) / 2 \text{ em outros casos.} \end{aligned}$$
- **Símbolos binários:** A medida de similaridade para símbolos binários é definida por:  $v(v_i, v_j) = 1$  se  $v_i = v_j$  e  $v(v_i, v_j) = 0$  caso contrário.
- **Símbolos ordenados:** A distância entre os valores de uma lista pode ser expressa pela associação de números de ranking para cada valor, assumindo uma ordem equidistante entre os valores, ou através de valores numéricos definidos pelo usuário. Com base nos valores

numéricos associados, as funções de similaridade descritas para números podem ser utilizadas para a determinação de valores de similaridade entre símbolos ordenados.

- **Símbolos não-ordenados:** Para a determinação de similaridade entre valores de uma lista finita de símbolos não ordenados, a medida de similaridade pode ser definida na forma de uma matriz de similaridade triangular  $S_{ij}$  que assinala cada combinação de valores  $v_i, v_j \in W(v)$  ao respectivo valor de similaridade  $v(v_i, v_j) \in [0, 1]$  como especificado em um ambiente específico.
- **Taxonomias:** Uma medida de similaridade para taxonomias é baseada na posição do objeto pela assinalação de valores de similaridade  $S_{v_i} \in [0, 1]$  para cada nó interno  $K_i$  da taxonomia, desde que atenda às seguintes condições: se  $K_j$  é o sucessor de  $K_i$  então  $S_{v_i} < S_{v_j}$ . Sendo  $K_i, K_j$  nós da taxonomia,  $\langle K_i, K_j \rangle$  o nó que é o predecessor mais próximo de  $K_i$  e  $K_j$ , e  $S_{V\langle K_i, K_j \rangle}$  o valor de similaridade assinalado ao nó  $\langle K_i, K_j \rangle$ . A medida de similaridade do nó  $K_i, K_j$  é então definida por  $v(K_i, K_j) = 1$  se  $K_i = K_j$ , ou  $v(v_i, v_j) = S_{V\langle K_i, K_j \rangle}$  nos outros casos.
- **String:** Uma abordagem para determinar a similaridade pode ser baseada na razão da maior parte comum [Ber+97]:  $v(v_i, v_j) = 0$  se  $|v_i| = 0$ ;  $v(v_i, v_j) = \frac{\sum |cv_j|}{\sum |w_i|}$  nos outros casos, sendo  $\text{Max } |cv_j|$  o comprimento da parte comum por palavra  $w_j$  de  $v_i \wedge (|cv_j| > 2) \wedge (|w_i| > 2)^1$ .

#### 5.4.1.5 Limite de similaridade

Quando uma base de experiências contém alguns poucos casos, como provavelmente no início da construção da mesma, muitos casos com uma similaridade muito baixa serão recuperados, devido a falta de alternativas mais apropriadas.

Para prevenir a sobrecarga do usuário com informação menos útil, casos com um valor de similaridade pequeno são excluídos pela definição de um limite. O limite demonstra explicitamente quando casos são considerados suficientemente similares.

Limites globais  $e_k \in [0,1]$  como parte da descrição do caso  $case_k = (E_k, e_k)$  define a borda inferior da qual o valor de similaridade global deve exceder para considerar este caso específico como uma candidato à reutilização.

O limite de similaridade local  $\theta_i \in [0,1]$  define para o índice  $Cg_i$  se dois valores são considerados como (suficientemente) similares.

Os limites podem ser atualizados com base no retorno do usuário sobre os candidatos à reutilização sugeridos.

#### 5.4.2 Medida de Similaridade para a Recuperação de experiências de Engenharia de Software

Considerando as necessidades específicas no domínio da engenharia de software, uma medida de similaridade  $\text{sim}(\text{Sit}', E_k')$  é definida. A medida de similaridade é baseada nas seguintes definições:

- Uma meta específica de recuperação  $g$ .
- Um conjunto de índices  $Cg = \{Cg_1, Cg_2, \dots\}$  para a meta de recuperação  $g$ .
- $\text{Sit}' = \{(Cg_i, s_i) \in \text{Sit} \mid \text{fator de relevância } (S_i) \neq \text{essencial}\}$ ,
- $\text{Case}_k = (E_k, e_k)$  dos dados da base com  $E_k' \subseteq E_k \wedge \forall E_{ki}' \in Cg \wedge$  características  $E_{ki}' \in Cg$  e seus respectivos valores  $e_{ki}'$ ,
- Valor de similaridade local  $v'(s_i, e_{ki}')$  dos valores  $s_i$  e  $e_{ki}'$ ,
- Peso  $\omega_{g_i}$  do vetor de relevância  $R_g$  da meta de recuperação  $g$ .
- Limite local  $\theta_i \in [0,1]$  para o índice  $Cg_i$ ,
- O conjunto de características com os pesos  $\alpha, \beta, \gamma, \delta \in [0,1]$ :  $E = \{Cg_i \mid (Cg_i \in \text{Sit}' \cap E_{ki}') \text{ e } (v'(S_i, e_{ki}') \geq \theta_i)\}$ ;  $W = \{Cg_i \mid (Cg_i \in \text{Sit}' \cap E_{ki}') \text{ e } (v'(S_i, e_{ki}') < \theta_i)\}$ ;  $U = \{Cg_i \mid (Cg_i \in E_{ki}' - \text{Sit}')\}$ ;  $R = \{Cg_i \mid (Cg_i \in \text{Sit}' - E_{ki}')\}$ .

A medida de similaridade global é definida por  $\text{sim}(\text{Sit}', E_k') = (\alpha \sum_{s_i \in E} \omega_{ik} v'(s_i, e_{ki}'))$

$$/ ((\alpha \sum_{s_i \in E} \omega_{ik} v'(s_i, e_{ki})) + (\beta \sum_{s_i \in W} \omega_{ik} (1-v'(s_i, e_{ki}))) + (\gamma \sum_{s_i \in U} \omega_{ik} (1-v'(s_i, e_{ki}))) + (\beta \sum_{s_i \in R} \omega_{ik} (1-v'(s_i, e_{ki}))))).$$

Case<sub>k</sub> é considerado como o candidato a reutilização, se todas as características marcadas como essenciais na situação dada combinarem exatamente com as respectivas características do caso e  $\text{sim}(\text{Sit}', E_k') \geq \text{limite global de similaridade } e_k \text{ de case}_k$ .

## **A FERRAMENTA REMEX**

A ferramenta REMEX é um ambiente abrangente que provê suporte para todo o processo de planejamento GQM. Suas funcionalidades incluem a edição e documentação de produtos GQM.

Mais adiante, ela provê facilidades para a reutilização de produtos GQM, assim como lições aprendidas de mensuração para todas as tarefas do processo de planejamento.

A ferramenta possui editores específicos para cada passo do processo de planejamento GQM de acordo com os produtos em particular a serem desenvolvidos.

Sempre que possível, a ferramenta suporta a criação semi-automática de produtos GQM com base nos produtos desenvolvidos nos passos anteriores do processo de planejamento, gerando relatórios de todos os produtos GQM relacionados ao processo de planejamento de acordo com a documentação definida para programas GQM.

Integradas à ferramenta REMEX estão facilidades para a reutilização de experiências de mensuração. Experiências de mensuração, assim como, conhecimento geral do domínio é armazenado na GQM-EB. A camada da aplicação provê ferramentas para a recuperação baseada em similaridade e orientada a meta de experiências de mensuração e a aquisição e integração de novas experiências. Ferramentas de busca e navegação estão disponíveis, assim como, ferramentas de manutenção permitindo ao engenheiro de conhecimento a integração de novas experiências na GQM-EB existente,

como será demonstrado a seguir.

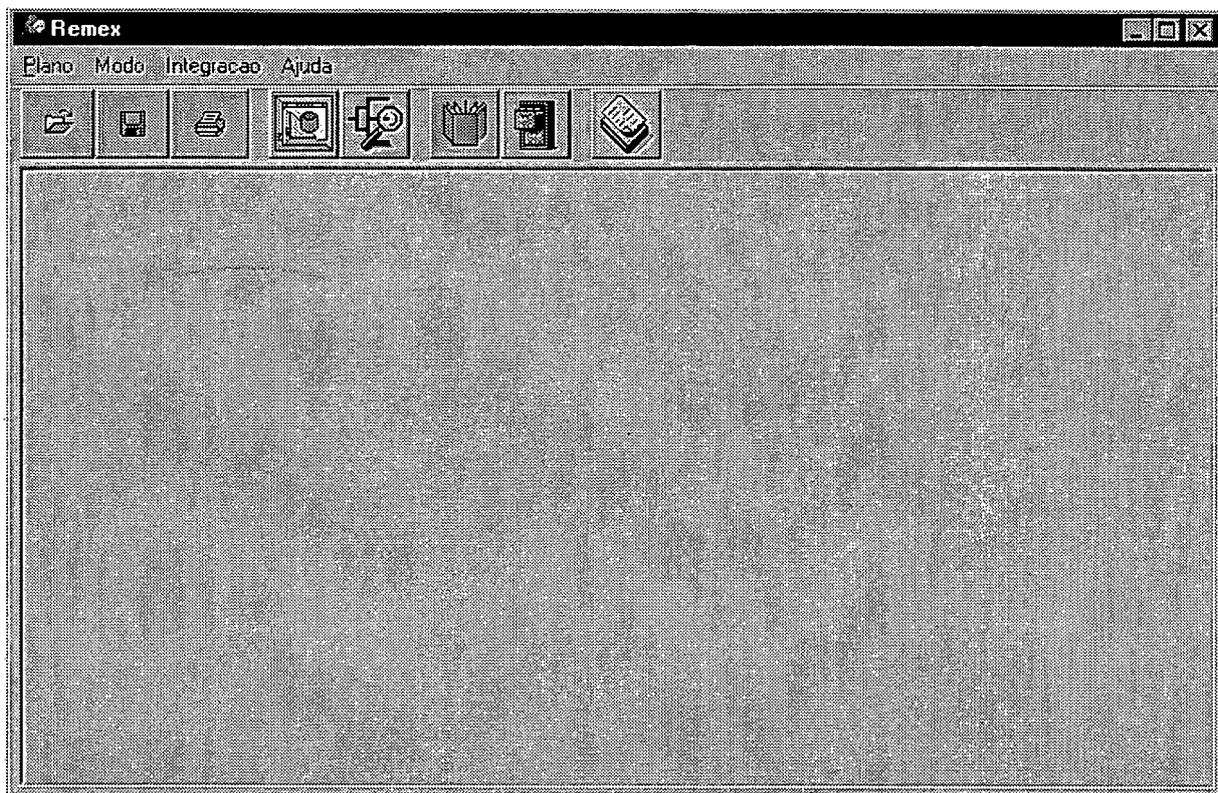


FIGURA 11. TELA PRINCIPAL DA FERRAMENTA REMEX

A tela principal da ferramenta (ver figura 11) dá acesso a todas os módulos que a compõe, provendo também atalhos para as funções mais comuns como abrir, salvar e imprimir planos de mensuração.

**Conhecimento Geral do Domínio:** O conhecimento, a respeito do domínio de aplicação dos planos de mensuração pode ser armazenado neste módulo, de maneira a facilitar o entendimento de estruturas, organizacionais, terminologias e outras informações que se fizerem necessárias.

**Prevenção de Problemas:** Durante o processo de planejamento, problemas poderão ocorrer, a função deste módulo é auxiliar o engenheiro de qualidade na prevenção destes, utilizando para isso a experiência adquirida em planejamentos já realizados.

**Resolução de Problemas:** Para problemas que ocorram, durante o processo de planejamento, este módulo auxiliará o engenheiro de qualidade, na resolução dos mesmos, também se utilizando o conhecimento adquirido em planejamentos passados.

**Base de Experiências:** A base de experiências (GQM-EB) é o repositório de

experiências adquiridas durante os planejamentos de mensuração. Neste módulo o engenheiro de conhecimento fará a manutenção dos casos existentes na GQM-EB, bem como das metas de recuperação de experiências do sistema.

## 1. Edição/Planejamento de um programa de Mensuração

Neste modulo é realizado todo o processo de construção dos planos de mensuração, de forma intuitiva e gradual, guiando o engenheiro de qualidade na construção dos referidos planos.

### 1.1 Caracterização do Contexto

Ao iniciarmos um novo plano de mensuração, o primeiro passo é construção de uma caracterização do ambiente de aplicação do mesmo. Na caracterização do contexto de aplicação do plano de mensuração serão identificadas as características da organização de software, do projeto alvo do programa de mensuração e do projeto de mensuração de forma a identificar fatores relevantes, que identifiquem o respectivo plano de mensuração.

### 1.1.1 Caracterização da Organização

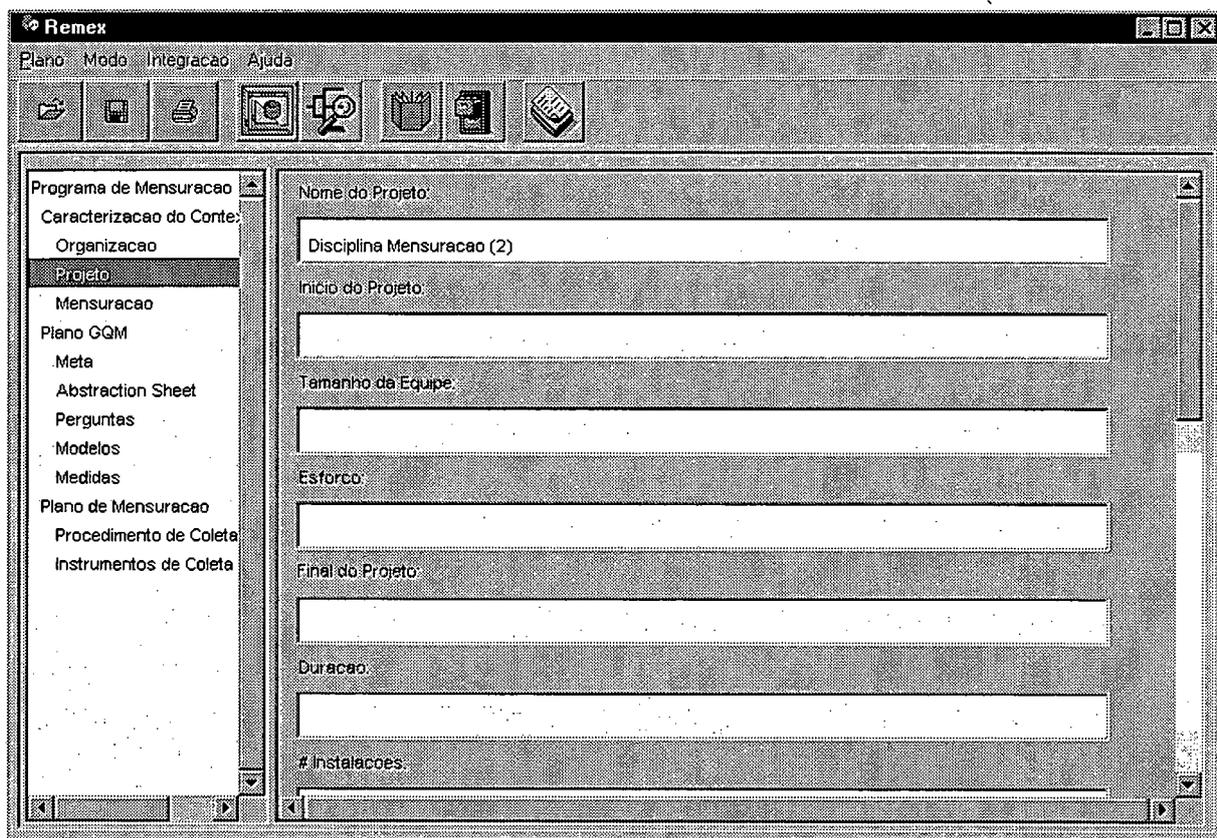
The screenshot displays the 'Remex' application window. The title bar shows 'Remex' and standard window controls. The menu bar contains 'Plano', 'Modo', 'Integracao', and 'Ajuda'. Below the menu bar is a toolbar with several icons. On the left, a vertical menu lists the following items: 'Programa de Mensuracao', 'Caracterizacao do Conte...', 'Organizacao' (highlighted), 'Projeto', 'Mensuracao', 'Plano GQM', 'Meta', 'Abstraction Sheet', 'Perguntas', 'Modelos', 'Medidas', 'Plano de Mensuracao', 'Procedimento de Coleta', and 'Instrumentos de Coleta'. The main workspace is divided into four sections: 'Nome da Organização' with a text box containing 'PPGEP'; 'Setor de Negocios' with an empty text box; 'Metas de Melhoramento' with an empty text box and vertical scroll arrows; and 'Comentarios' with an empty text box and vertical scroll arrows. A 'Gravar' button is positioned at the bottom right of the main workspace.

FIGURA 12. MODO PLANEJAMENTO: CARACTERIZAÇÃO DA ORGANIZAÇÃO

A primeira fase desta caracterização é a organização envolvida no processo de mensuração, que é identificada como exemplificado na figura 12.

Esta caracterização da organização compreende a identificação da mesma, sua área de atuação, e as metas de melhoramento que pretende atingir com o programa de qualidade.

### 1.1.2 Caracterização do Projeto



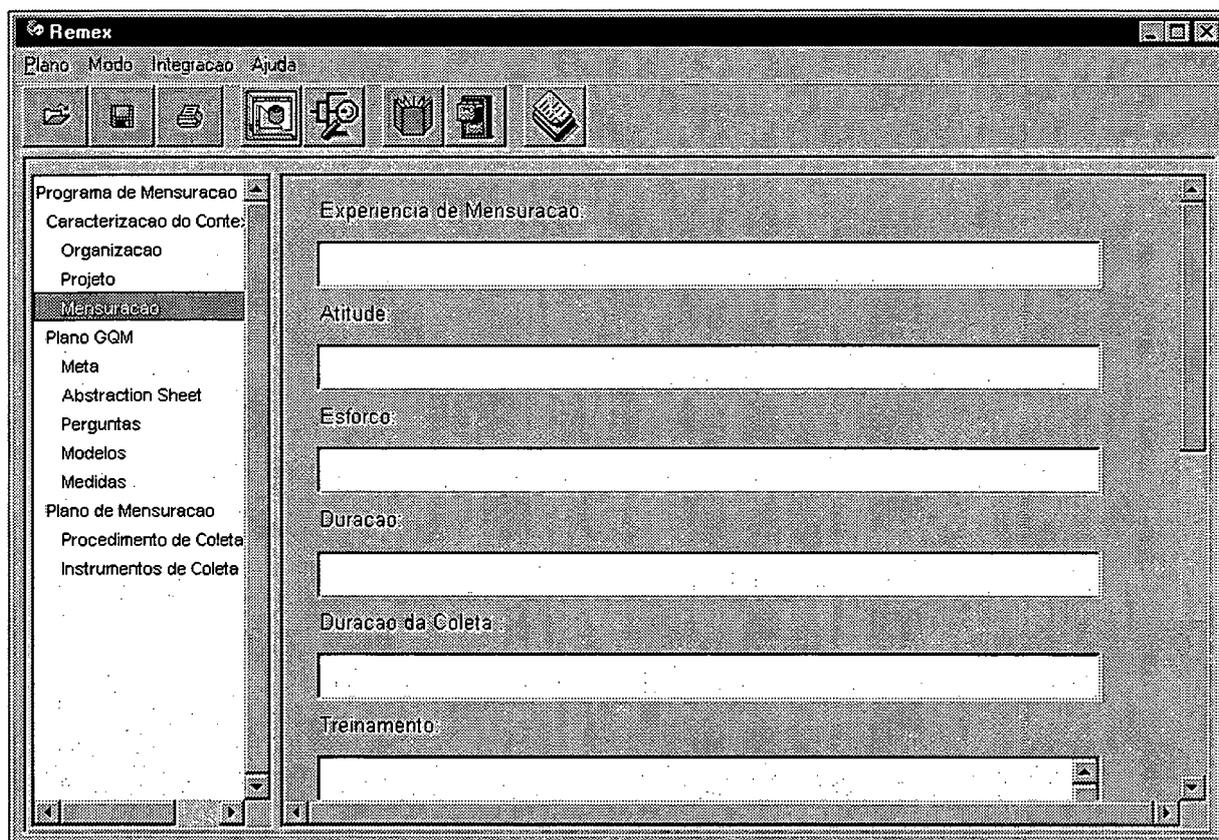
The screenshot shows the 'Remex' application window with the 'Plano' menu selected. The interface is divided into a left sidebar and a main content area. The sidebar contains a tree view with the following items: Programa de Mensuração, Caracterização do Conteúdo, Organização, Projeto (highlighted), Mensuração, Plano GQM, Meta, Abstraction Sheet, Perguntas, Modelos, Medidas, Plano de Mensuração, Procedimento de Coleta, and Instrumentos de Coleta. The main content area contains a form with the following fields: Nome do Projeto (with the text 'Disciplina Mensuração (2)'), Início do Projeto, Tamanho da Equipe, Esforço, Final do Projeto, Duração, and # Instalações.

FIGURA 13. MODO PLANEJAMENTO: CARACTERIZAÇÃO DO PROJETO

O segundo passo da caracterização do ambiente de aplicação do plano de mensuração é identificar o projeto de software no qual será aplicado o programa de qualidade.

Para tanto dados como nome do projeto, início, duração, equipem e linguagens de programação podem ser obtidas com o gerente ou responsáveis pela condução do projeto e armazenados no formulário demonstrado na figura 13.

### 1.1.3 Caracterização da Mensuração



The screenshot shows the 'Remex' software window. The title bar reads 'Remex'. The menu bar includes 'Plano', 'Modo', 'Integracao', and 'Ajuda'. Below the menu bar is a toolbar with icons for file operations and navigation. The main window is divided into two panes. The left pane is a tree view with the following structure:

- Programa de Mensuracao
  - Caracterizacao do Conte:
    - Organizacao
    - Projeto
    - Mensuracao (highlighted)
  - Plano GGM
    - Meta
    - Abstraction Sheet
    - Perguntas
    - Modelos
    - Medidas
  - Plano de Mensuracao
    - Procedimento de Coleta
    - Instrumentos de Coleta

The right pane is a form titled 'Experiencia de Mensuracao' with several input fields:

- Experiencia de Mensuracao: [Text Field]
- Atitude: [Text Field]
- Esforco: [Text Field]
- Duracao: [Text Field]
- Duracao da Coleta: [Text Field]
- Treinamento: [Text Field]

FIGURA 14. MODO PLANEJAMENTO: CARACTERIZAÇÃO DA MENSURAÇÃO

O terceiro e último passo na construção da caracterização do ambiente é constituído pela caracterização do processo de mensuração (ver figura 14) que será efetuado, a partir da definição de informações como a experiência da equipe de mensuração, o esforço envolvido, a duração do processo e da coleta de dados, bem como a necessidade do treinamento da equipe ou não, além de outras informações que se fizerem necessárias.

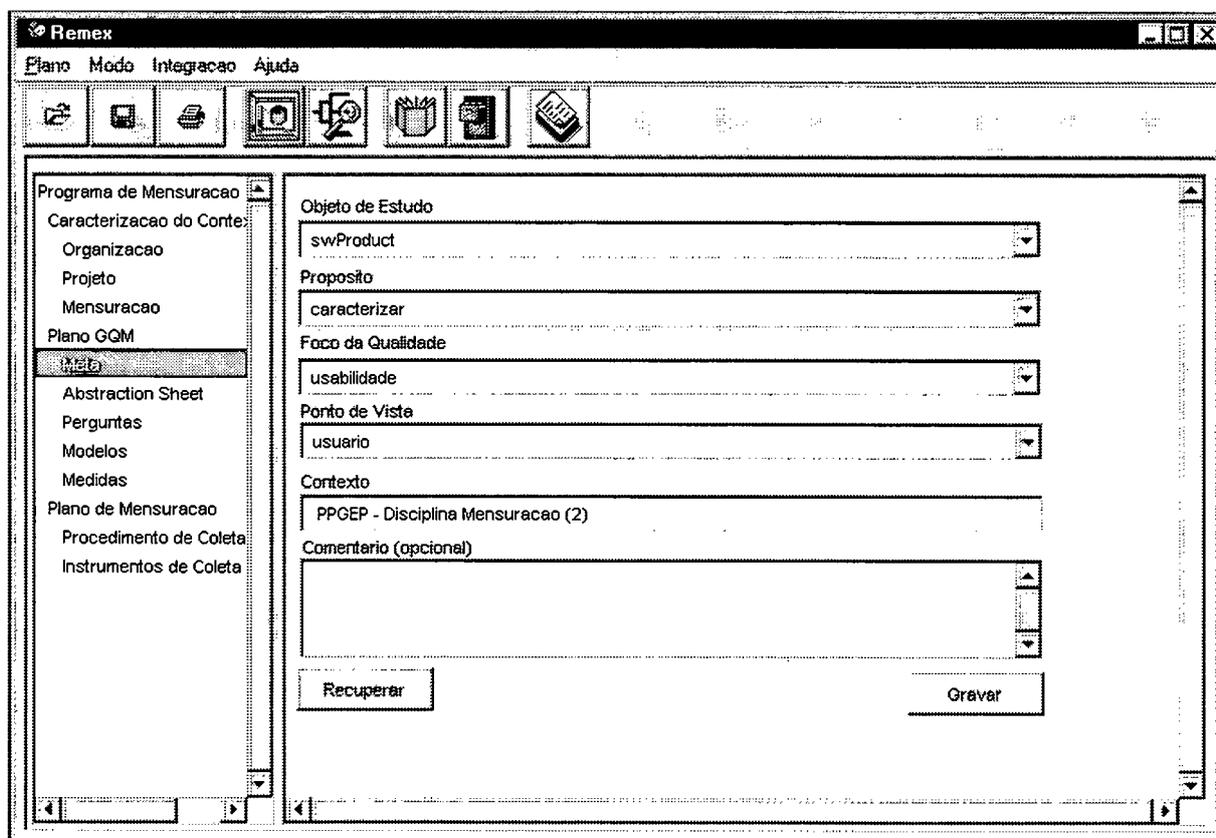
As informações aqui armazenadas são de grande importância uma vez que caracterizam o processo de mensuração, através de informações como a duração do processo de coleta de dados que determinará o tempo necessário para a coleta de amostras suficientes para a análise e obtenção de resultados válidos para o processo de qualidade.

## 1.2 Definição do Plano de Mensuração

O segundo grande passo no planejamento de um programa de mensuração baseado em GQM, é a elaboração das medidas e serem executadas na fase de coleta de dados, para isso os seguintes passos descritos a seguir necessitam ser seguidos.

### 1.2.1 Identificação da Meta de Mensuração

As dimensões da meta representam o foco de todo o processo, determinando o que será estudado, com que propósito, define o ponto de vista dos interessados nos resultados da mensuração, e identifica o contexto de aplicação do programa de mensuração.



The screenshot shows the 'Remex' application window. The menu bar includes 'Plano', 'Modo', 'Integracao', and 'Ajuda'. The toolbar contains icons for file operations and navigation. The left sidebar lists the application's structure, with 'Meta' selected under 'Plano GQM'. The main area is a form for defining a goal, with the following fields:

- Objeto de Estudo:** swProduct
- Proposito:** caracterizar
- Foco da Qualidade:** usabilidade
- Ponto de Vista:** usuario
- Contexto:** PPGEF - Disciplina Mensuracao (2)
- Comentario (opcional):** (empty text area)

At the bottom of the form are two buttons: 'Recuperar' and 'Gravar'.

FIGURA 15. MODO PLANEJAMENTO: DEFINIÇÃO DA META DE MENSURAÇÃO

As informações, como, por exemplo, objeto de estudo, foco de qualidade e ponto de vista são armazenados no formulário demonstrado na figura 15.

### 1.2.2 Execução das Entrevistas

Após a definição da meta do projeto de mensuração, o engenheiro de qualidade, conduzirá as entrevistas com as pessoas envolvidas no ponto de vista, a fim de coletar o conhecimento necessário à definição das dimensões de qualidade a serem buscadas no processo de mensuração.

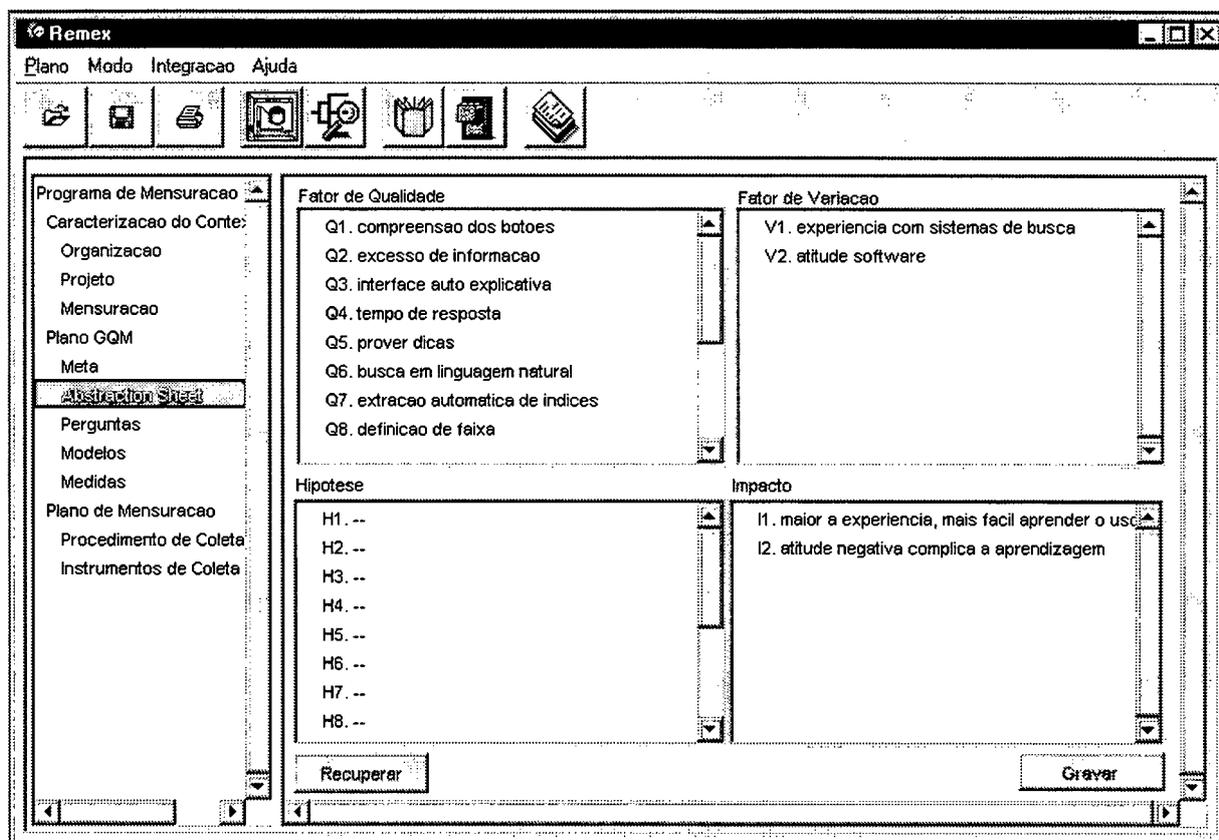


FIGURA 16. MODO PLANEJAMENTO: DEFINIÇÃO DO ABSTRACTION SHEET.

Para isso utilizará um instrumento chamado Abstraction Sheet, para cada pessoa entrevistada. Ao final do processo de aquisição do conhecimento, as informações de todos os Abstraction Sheet's serão reunidas e armazenadas no sistema (ver figura 16). É possível efetuar a edição de qualquer item do Abstraction Sheet, utilizando-se dos formulários demonstrados na figura 17.

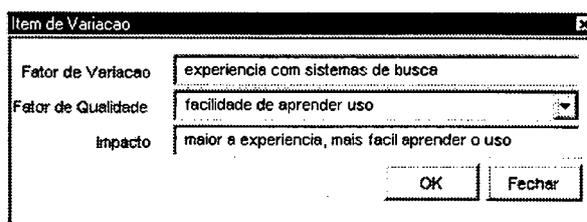


FIGURA 17. MODO PLANEJAMENTO: EDIÇÃO DOS ITENS DO ABSTRACTION SHEET

### 1.2.3 Definição das Perguntas GQM

Passada a definição do Abstraction Sheet, o engenheiro de qualidade deve refinar a informação adquirida através das entrevistas, transformando os fatores de qualidade, e variação em perguntas que futuramente serão respondidas.

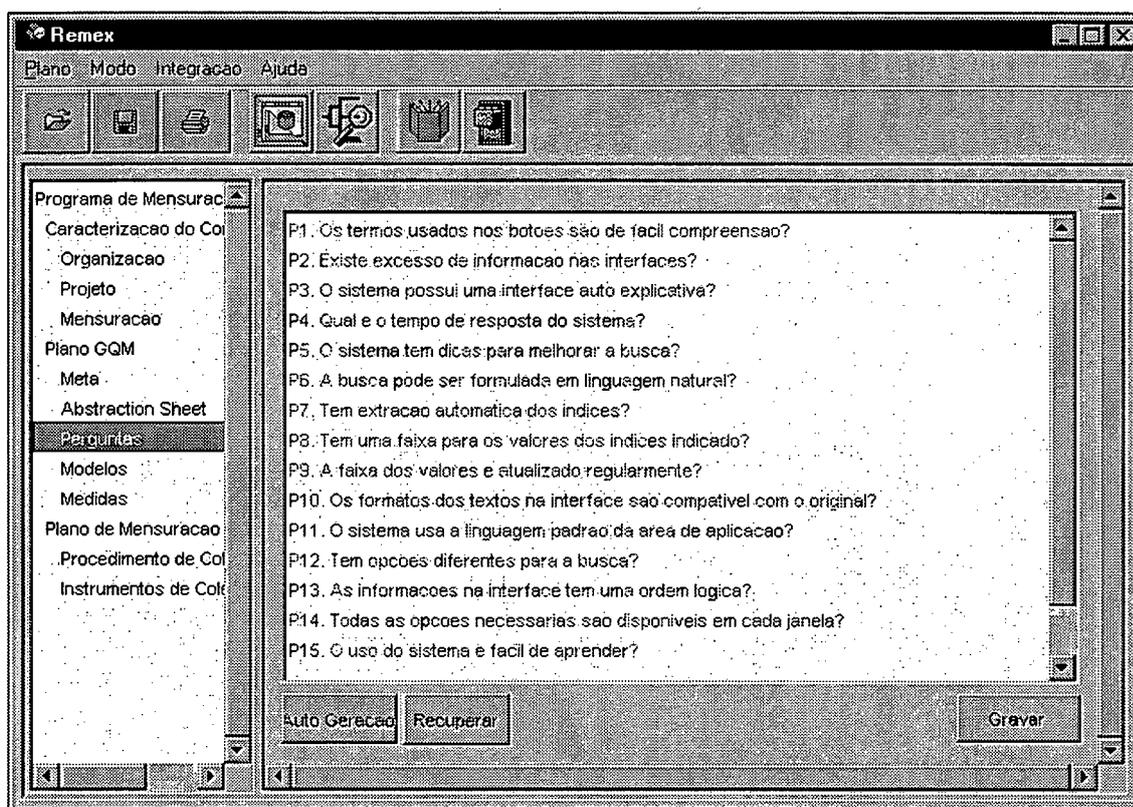
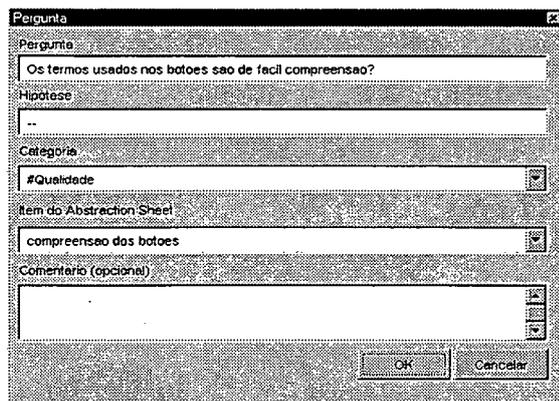


FIGURA 18. MODO PLANEJAMENTO: DEFINIÇÃO DAS PERGUNTAS

Para tanto a ferramenta possui o recurso de geração automática das perguntas, o



que permitirá ao engenheiro de qualidade concentrar-se no refinamento das perguntas, diminuindo o esforço envolvido nesta fase do planejamento (ver figura 18).

A definição de cada pergunta é realizada através de um formulário próprio, onde será definida a pergunta em si, uma hipótese para a mesma, sua categoria e uma

referência ao item do Abstraction Sheet que a originou (ver figura 19).

FIGURA 19. MODO PLANEJAMENTO: EDIÇÃO DE PERGUNTAS

### 1.2.4 Definição dos Modelos de Qualidade

Ao iniciar a definição dos modelos, uma relação derivada de todas as perguntas definidas na etapa anterior, é automaticamente gerada e apresentada ao engenheiro de qualidade (ver figura 20), o qual tem a possibilidade de manipulá-los a partir da lista apresentada, sendo possível editar em um formulário próprio o modelo definido para cada pergunta.

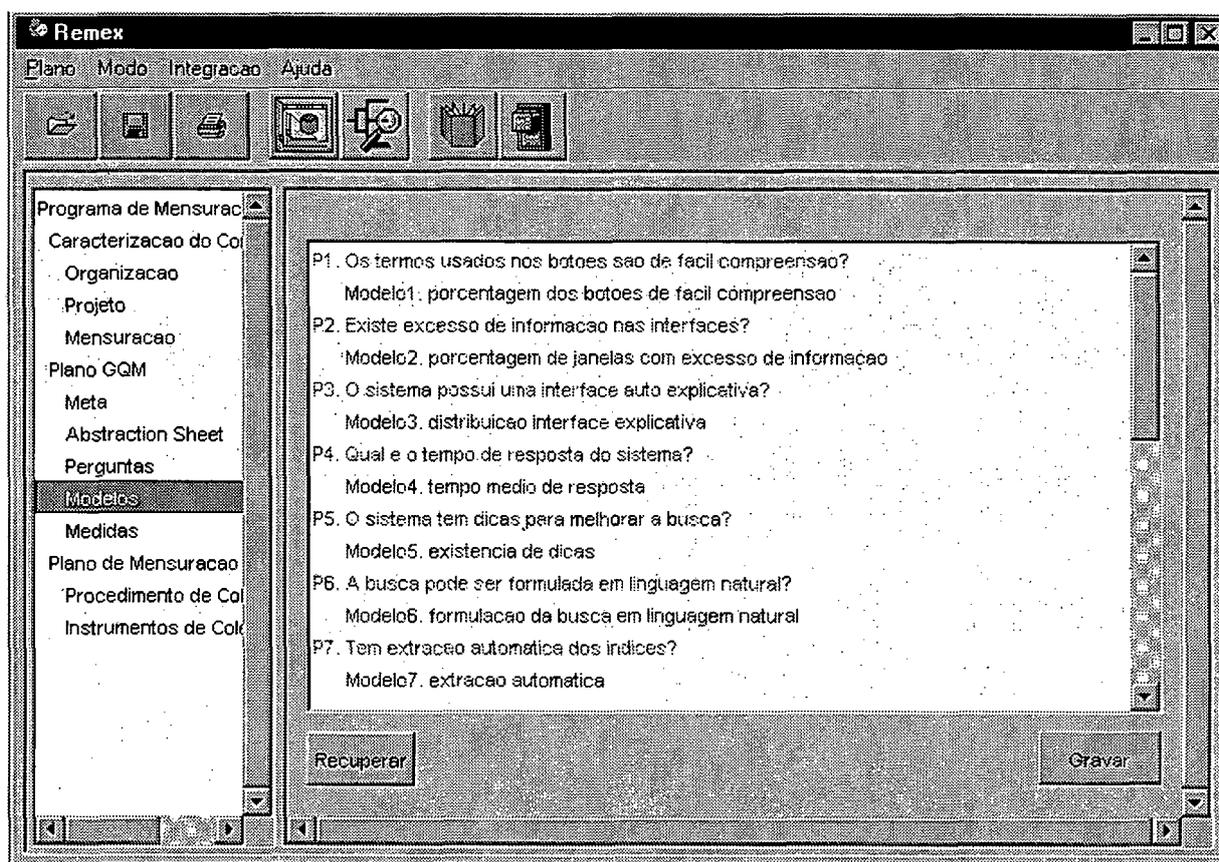


FIGURA 20. MODO PLANEJAMENTO: DEFINIÇÃO DOS MODELOS DE QUALIDADE

Na edição dos modelos de qualidade, todas as informações referentes à formalização da pergunta, refinando-a e definindo-a de uma forma quantitativa e mensurável, é armazenada em um formulário próprio. Este formulário (ver figura 21), guiará

o engenheiro de qualidade na definição de suas informações, diminuindo o esforço e a possibilidade de que se cometam erros em sua definição.

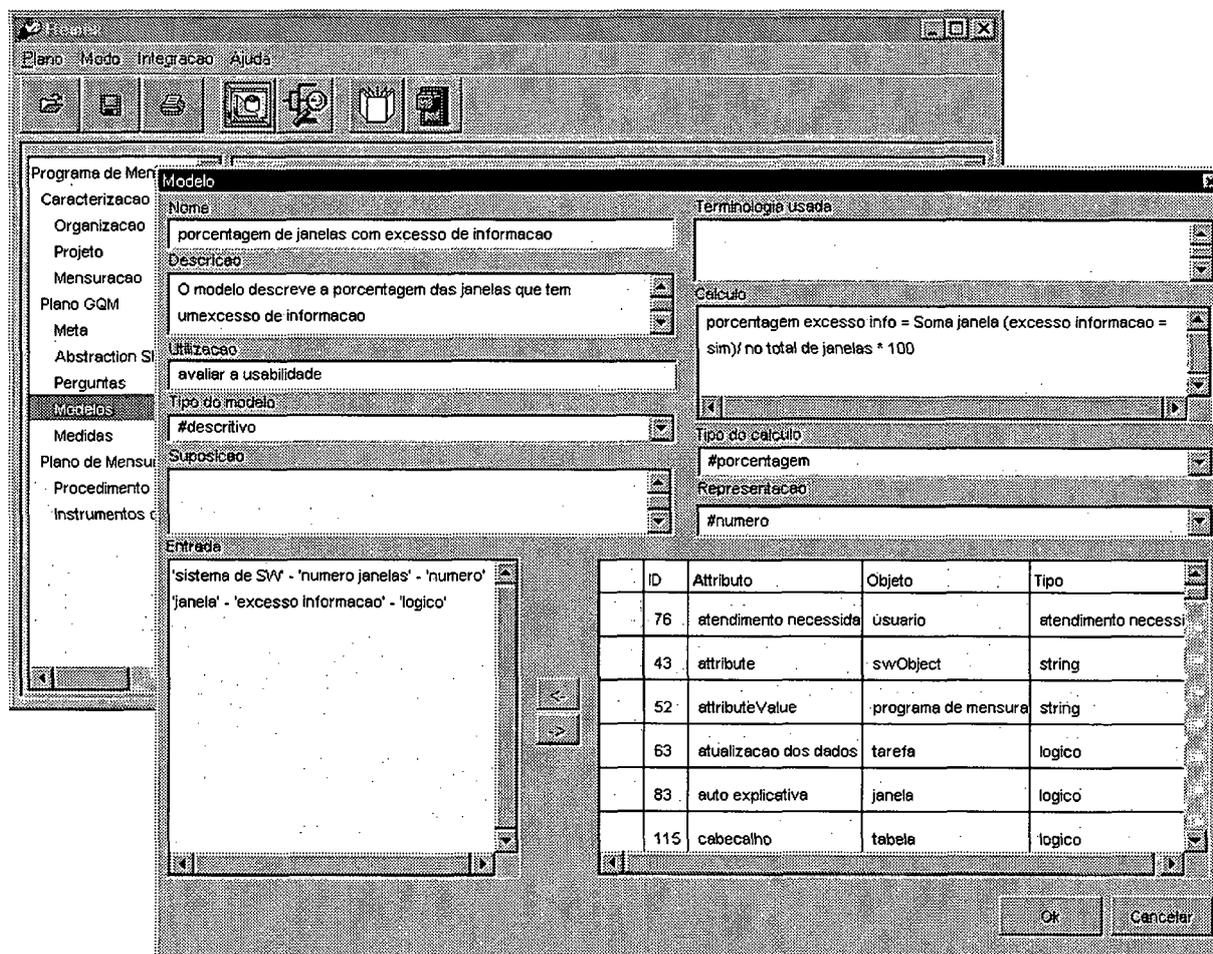


FIGURA 21. MODO PLANEJAMENTO: EDIÇÃO DE MODELOS DE QUALIDADE

Informações tais como utilização, tipo de modelo, tipo de cálculo, forma de representação do resultado, bem como dos atributos envolvidos na formulação do cálculo do modelo serão definidas no formulário de edição e posteriormente armazenadas junto da respectiva pergunta do atual plano de mensuração.

### 1.2.5 Definição das Medidas GQM

O passo seguinte constitui na transformação dos atributos definidos como

entradas do modelo de qualidade em medidas a serem executadas na fase de coleta de dados. Para facilitar esta tarefa o sistema gera automaticamente uma pré-definição destas medidas com base nos atributos de entrada dos modelos de qualidade definidos no passo anterior.

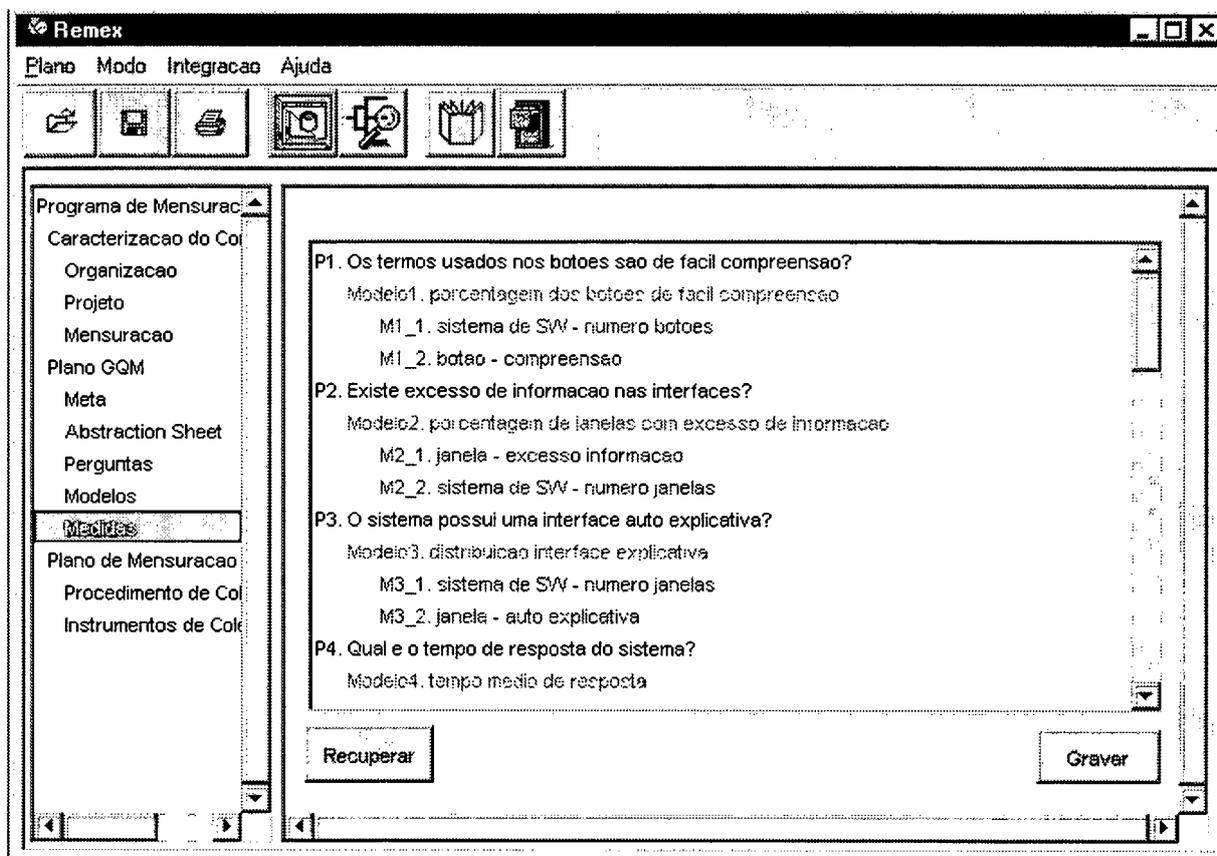
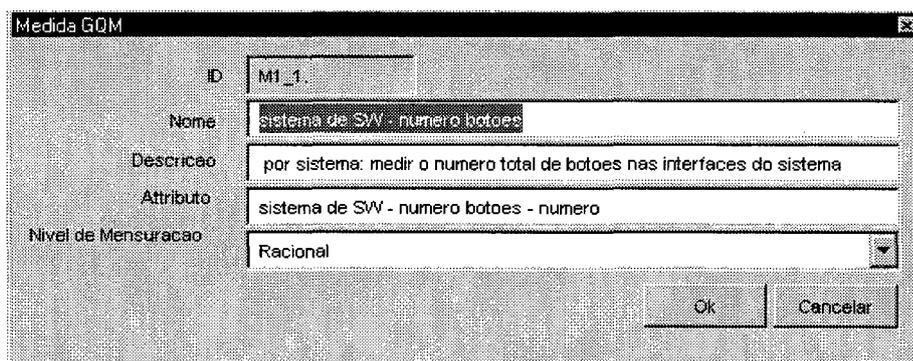


FIGURA 22. MODO PLANEJAMENTO: DEFINIÇÃO DAS MEDIDAS.

Uma listagem contendo as perguntas, seu modelo e as medidas derivadas dos atributos dos modelos serão apresentados (ver figura 22), para que o engenheiro de qualidade possa refinar a sua definição, através da edição de cada uma das medidas geradas, ou da recuperação e posterior adequação de medidas similares encontradas na GQM-EB.



The image shows a dialog box titled "Medida GQM" with the following fields:

ID	MI_1.
Nome	sistema de SW - numero botoes
Descrição	por sistema: medir o numero total de botoes nas interfaces do sistema
Atributo	sistema de SW - numero botoes - numero
Nivel de Mensuracao	Racional

Buttons: Ok, Cancelar

FIGURA 23. MODO PLANEJAMENTO: EDIÇÃO DE MEDIDAS

Cada medida pode, então ter suas informações editadas (ver figura 23), de forma a ser definido uma identificação, uma descrição explicando a medida, o atributo a ser medido e o nível de mensuração a ser realizado.

### 1.3 Definição do Plano de Mensuração.

O terceiro grande passo na elaboração de um processo de mensuração baseado em GQM é a elaboração do plano de mensuração, que definirá quando os dados serão coletados, que dados serão coletados, por quem estes dados serão coletados e de que forma eles serão coletados.

### 1.3.1 Procedimentos de Coleta de Dados

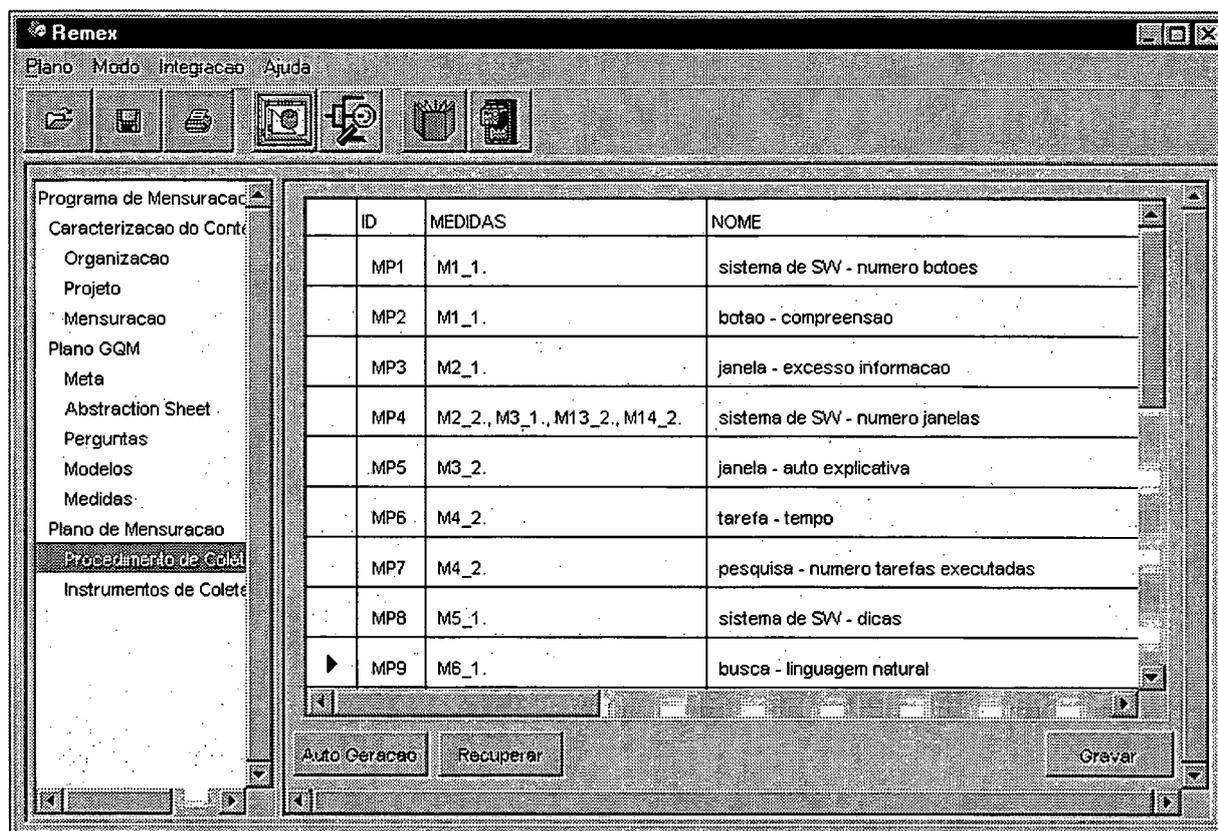


FIGURA 24. MODO PLANEJAMENTO: DEFINIÇÃO DOS PROCEDIMENTOS DE MENSURAÇÃO

Para auxiliar no processo de definição dos procedimentos de coleta de dados a ferramenta gera uma lista unificada de todas as medidas (ver figura 24), reunindo as medidas idênticas em um único procedimento de coleta, para que o esforço envolvido na coleta de dados seja minimizado, otimizando assim o processo de coleta.

### 1.3.2 Instrumentos de Coleta de Dados

Com base nas informações obtidas a partir dos modelos, medidas e procedimentos de coleta de dados, a ferramenta gera algumas sugestões de instrumentos de coleta de dados (ver figura 25) que podem ser utilizados como base para a elaboração definitiva dos instrumentos a serem utilizados na coleta, reduzindo desta forma o esforço envolvido na elaboração dos formulários, entrevistas ou questionários a serem aplicados durante o processo de mensuração.

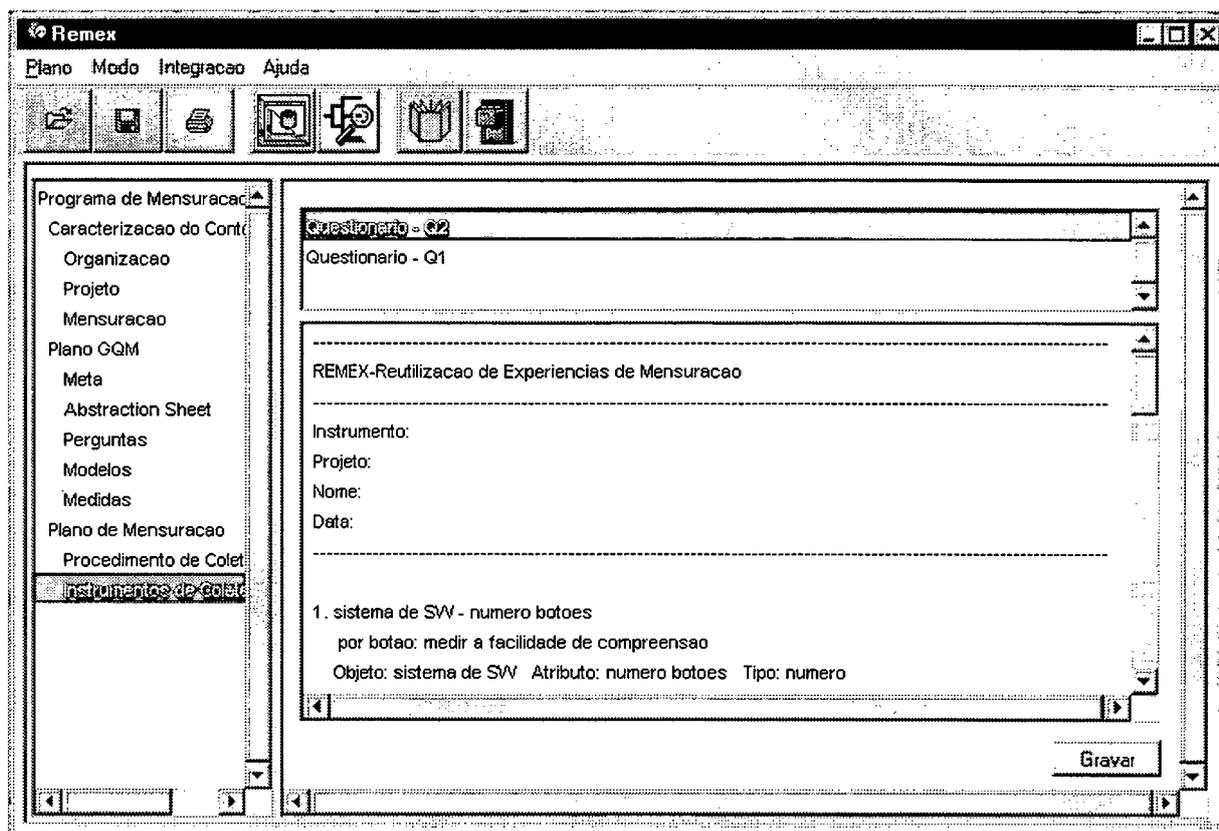


FIGURA 25. MODO PLANEJAMENTO: DEFINIÇÃO DOS INSTRUMENTOS DE MENSURAÇÃO

## 2. Recuperação de Experiências

### 2.1 Recuperação de PEC

A partir do início do processo de planejamento, já é possível ao engenheiro de qualidade a utilização do conhecimento armazenado na GQM-EB, para a melhor definição dos produtos GQM. Para tanto, o sistema utilizará as informações armazenadas nas fases anteriores à do produto GQM em desenvolvimento para a identificação de produtos que tenham características similares às do atual projeto.

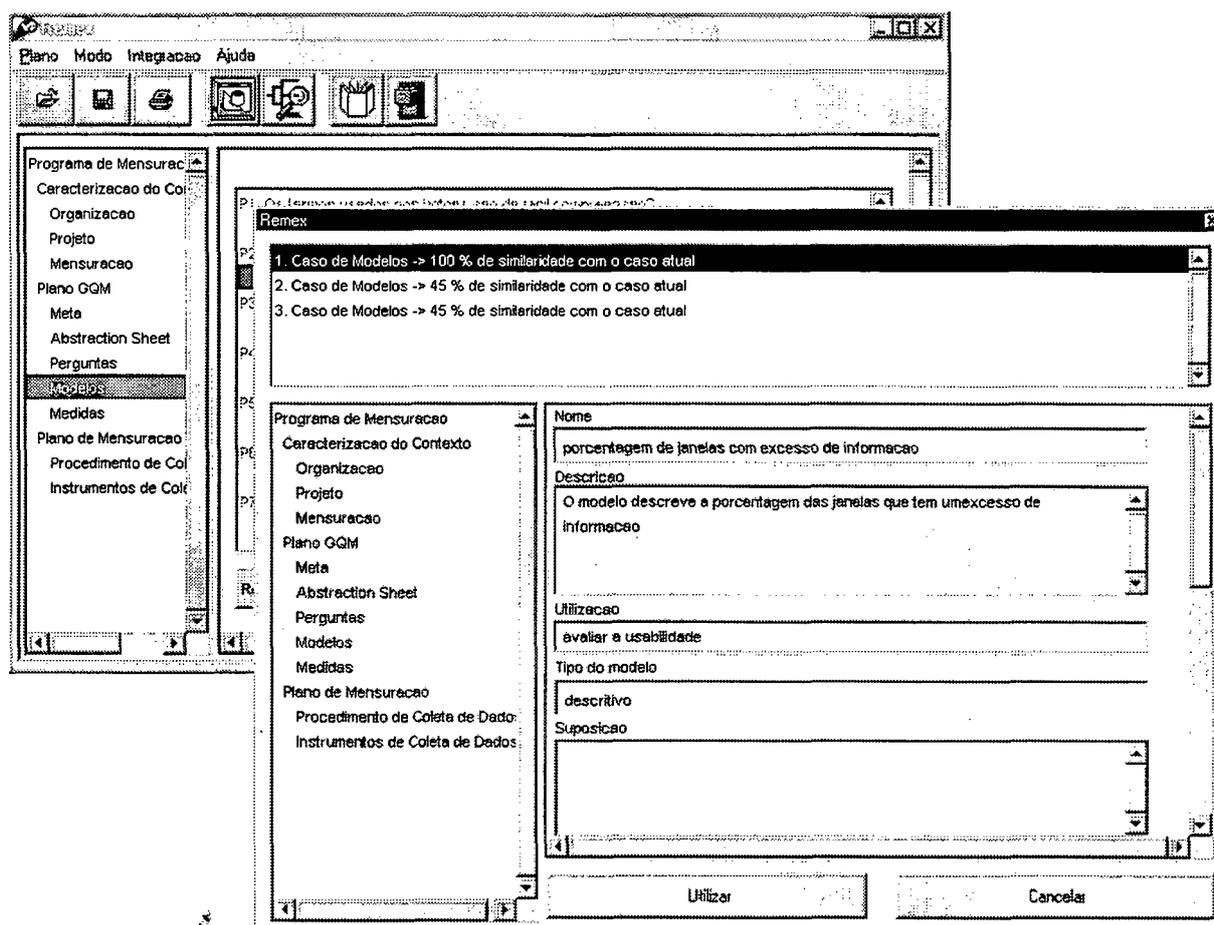


FIGURA 26. RECUPERAÇÃO DE EXPERIÊNCIAS: MODELOS DE QUALIDADE.

O engenheiro de qualidade pode, a partir do formulário de recuperação demonstrado na figura 26, selecionar o caso mais apropriado e utilizá-lo da definição do produto GQM em desenvolvimento no projeto atual, ou ainda se achar necessário, navegar por todo o caso apresentado na interface de recuperação a fim de certificar-se da validade do caso apresentado.

## 2.2 Recuperação de PSEC

A recuperação de experiências de problemas e soluções permite ao engenheiro de qualidade utilizar o conhecimento adquirido durante a resolução de problemas que ocorreram em planejamentos passado, diminuindo assim o esforço na resolução ou prevenção de problemas que possam ocorrer durante as fases do planejamento do plano GQM.

### 2.2.1 Prevenção de Problemas

A prevenção de problemas é uma facilidade que permite ao engenheiro de qualidade pode utilizar as experiências adquiridas na resolução de problemas, durante os planejamentos de mensuração passados para precaver-se quanto à ocorrência de problemas em cada nova fase do planejamento. Para isto, o modo de prevenção de problemas oferece um formulário no qual é possível escolher qual a fase do processo de planejamento se deseja buscar por problemas que por ventura possam ocorrer.

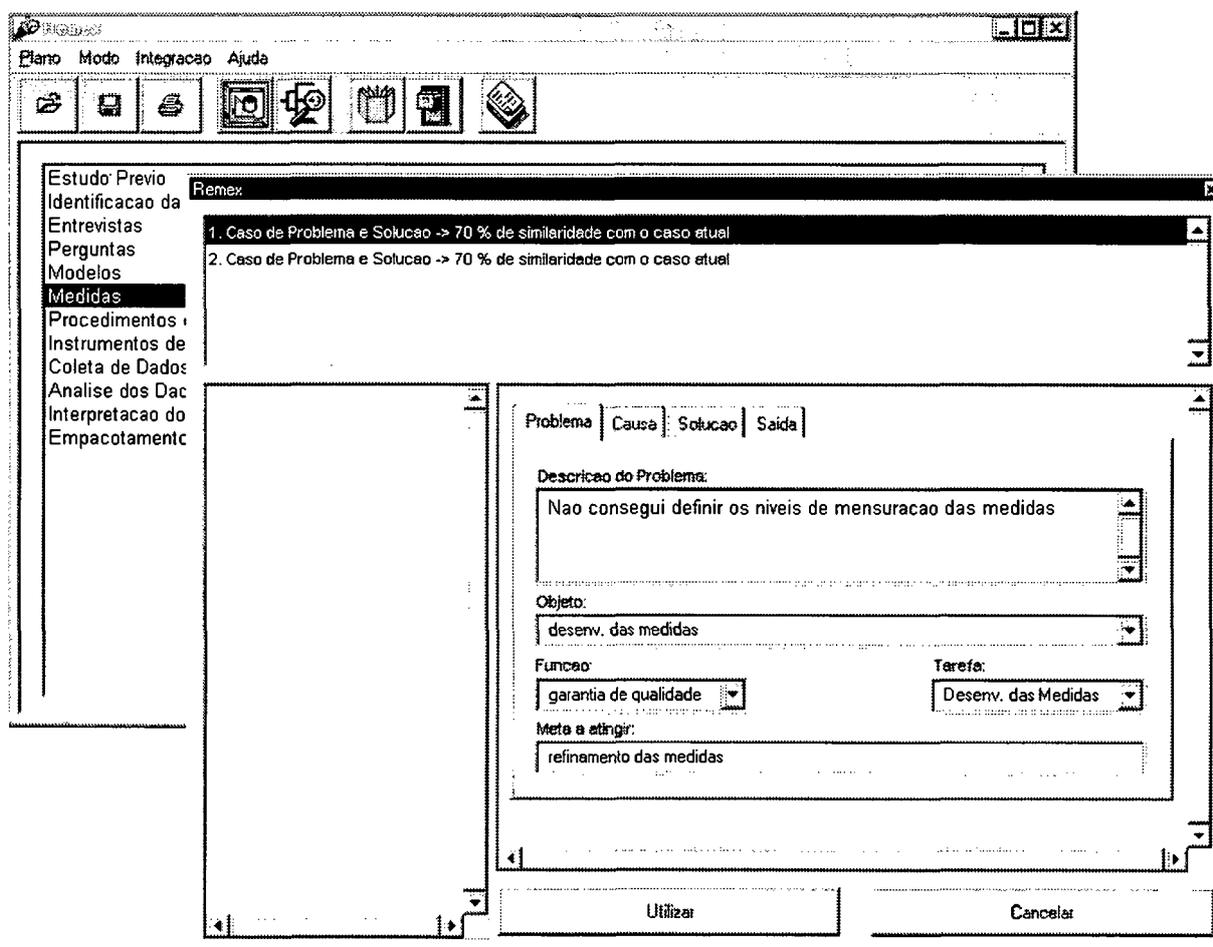


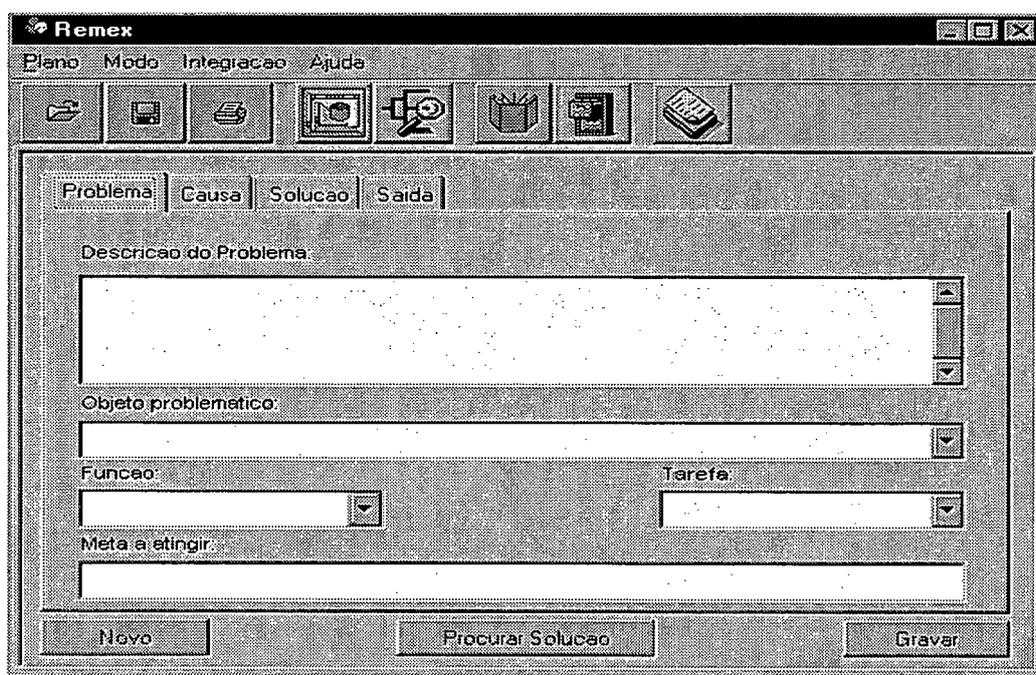
FIGURA 27. RECUPERAÇÃO DE EXPERIÊNCIAS: PREVENÇÃO DE PROBLEMAS

Quando da seleção de uma fase do processo de planejamento de mensuração, o sistema automaticamente busca na GQM-EB por problemas e soluções ocorridos nesta

fase do planejamento em planos de mensuração executados no passado, e os apresenta ao engenheiro de qualidade (ver figura 27), para que este possa analisá-los buscando situações que possam ocorrer, na fase desejada, e verificando as ações que foram tomadas a fim de sanar as dificuldades encontradas, agindo de forma preventiva no delineamento do plano de mensuração.

### 2.2.2 Resolução de Problemas

No modo de resolução de problemas, o engenheiro de qualidade pode utilizar-se do conhecimento armazenado na GQM-EB, para solucionar problemas que não puderam ser previstos, através da descrição do problema no respectivo formulário (ver figura 37) e da identificação de sua causa se isso for possível, podendo então solicitar ao sistema que procure por situações similares que tenham ocorrido em programas de mensuração passados.



The image shows a screenshot of the Remex software interface. The window title is "Remex". The menu bar includes "Plano", "Modo", "Integracao", and "Ajuda". The toolbar contains several icons for file operations and problem management. Below the toolbar, there are four tabs: "Problema", "Causa", "Solucao", and "Saida", with "Problema" selected. The main area contains a "Descricao do Problema" text area, an "Objeto problematico" dropdown menu, a "Funcao" dropdown menu, a "Tarefa" dropdown menu, and a "Meta a atingir" text area. At the bottom, there are three buttons: "Novo", "Procurar Solucao", and "Gravar".

FIGURA 28. RECUPERAÇÃO DE EXPERIÊNCIAS: DESCRIÇÃO DE UM NOVO PROBLEMA

Após a descrição do problema e sua causa quando possível, o sistema apresenta ao engenheiro de qualidade os casos mais similares encontrados na GQM-EB (ver figura 29), para que este possa navegar através dos itens apresentados buscando a solução mais adequada à solução do problema anteriormente descrito.

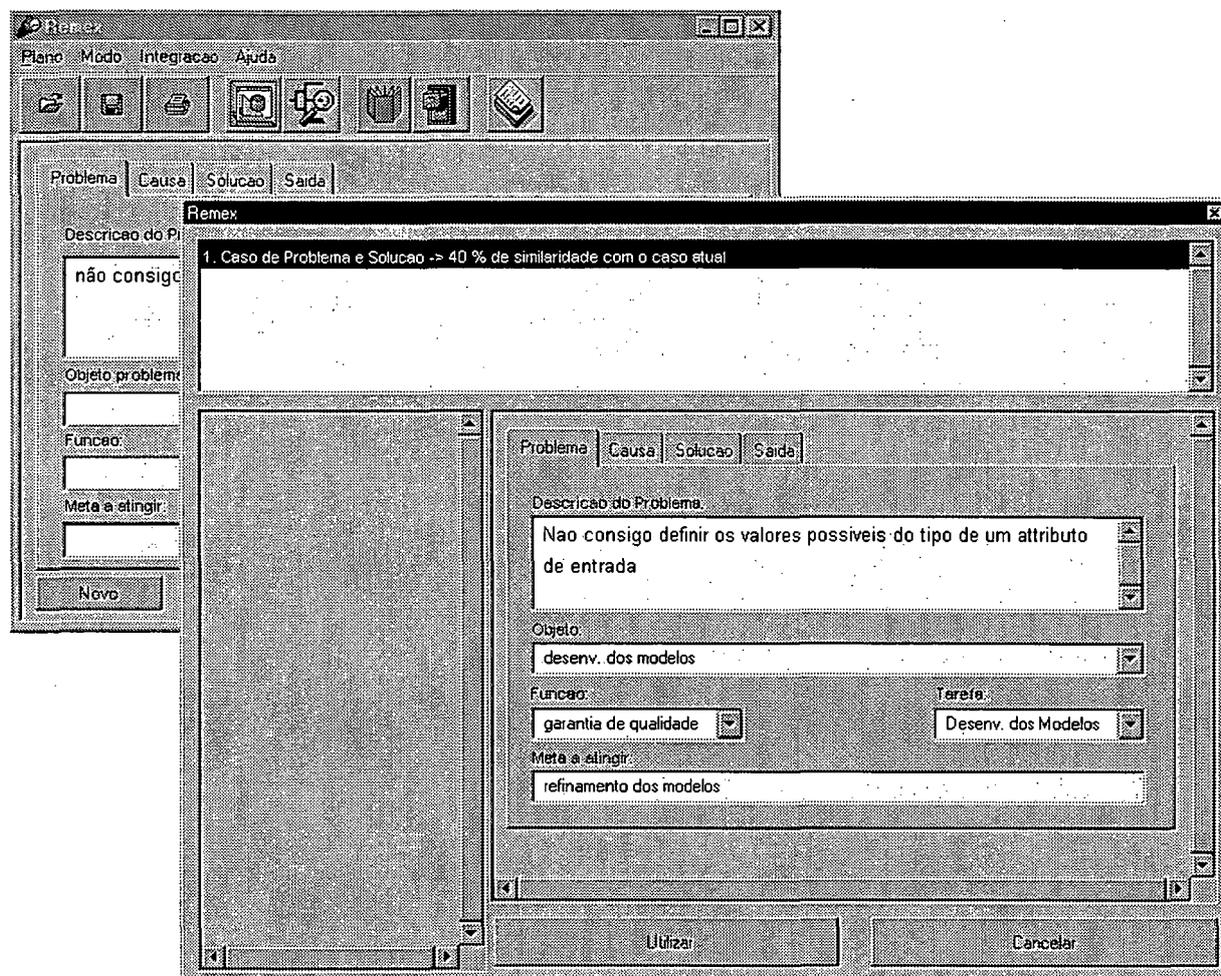


FIGURA 29. RECUPERAÇÃO DE EXPERIÊNCIAS: RECUPERAÇÃO DE SOLUÇÕES

Se optar por utilizar uma das soluções sugeridas, o engenheiro de qualidade, pode então adaptar a solução, se isto se fizer necessário, registrando também o resultado da aplicação da solução para futuras consultas.

### 3. Aquisição do Conhecimento

A aquisição do conhecimento permite que toda a informação obtida durante o planejamento seja armazenada na GQM-EB, formando assim um repositório de casos que estarão disponíveis por toda a organização para serem utilizados na construção de futuros planos de mensuração.

### 3.1 Aquisição de Casos de PEC's.

Quando um plano de mensuração é encerrado, passa a fazer parte de um repositório de casos a serem adicionados na GQM-EB (ver figura 30), o que significa que não se tornam imediatamente disponíveis uma vez que deverão passar pela avaliação do engenheiro de conhecimento, que pode aceitar ou rejeitar o caso se considera-lo como uma experiência irrelevante.

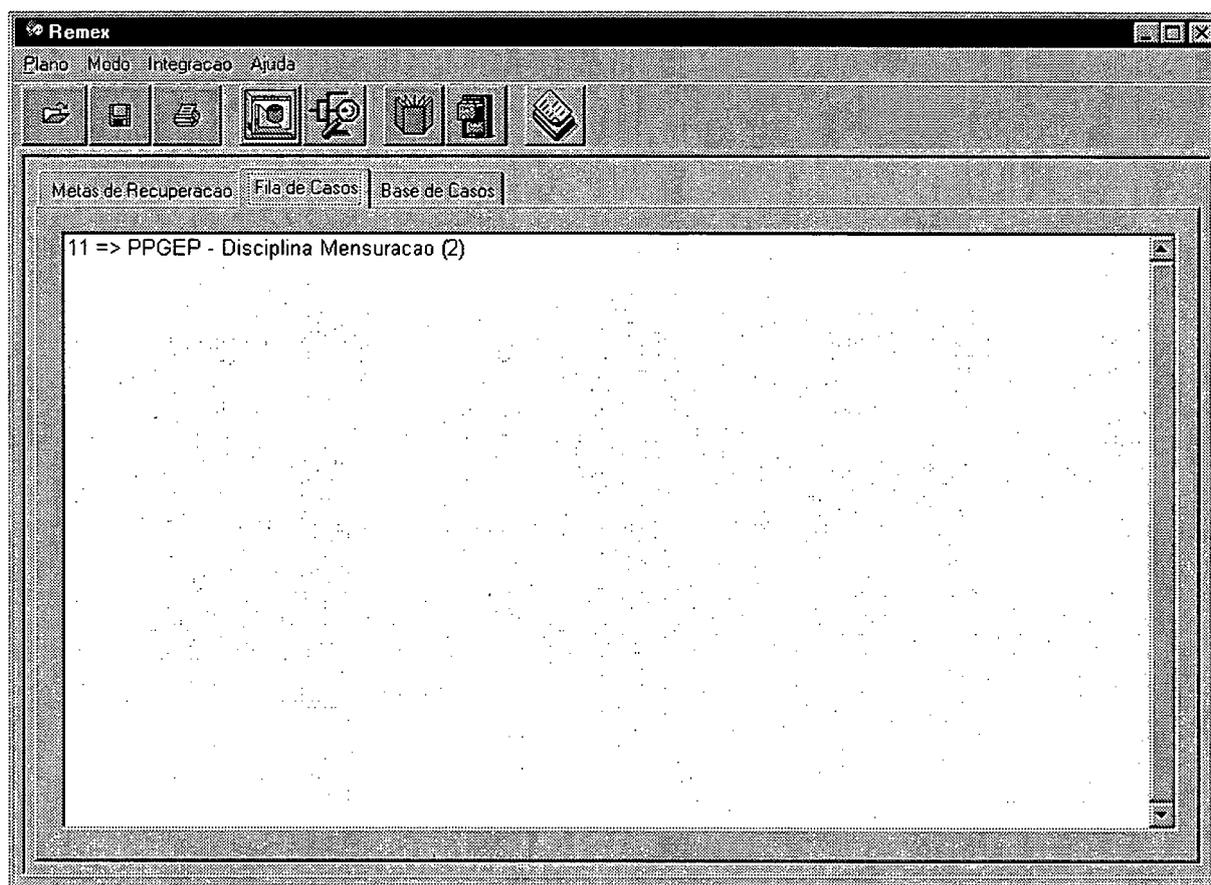


FIGURA 30. AQUISIÇÃO DE CONHECIMENTO: FILA DE CASOS A SEREM ADICIONADOS NA GQM-EB

Após a análise e aprovação do caso pelo engenheiro de conhecimento, ele torna-se disponível na GQM-EB para que futuros planejamentos de mensuração possam utilizar o conhecimento adquirido durante o seu planejamento.

### 3.2 Aquisição de PSEC's.

A aquisição de experiências de Problemas e Soluções (PSEC), é realizada automaticamente pela ferramenta, uma vez que a cada problema resolvido, o conhecimento armazenado no caso PSEC é transferido para a GQM-EB e tornando-se automaticamente disponível, possibilitando a rápida disseminação do conhecimento por toda a organização de software.

## 4. Manutenção do Sistema

O modo de manutenção permite ao engenheiro de qualidade o ajuste e adequação da ferramenta à organização específica, de forma a possibilitar o desenvolvimento dos planos de mensuração e a recuperação de conhecimento da GQM-EB da forma mais adequada possível.

### 4.1 Hierarquia de Objetos

No modo de conhecimento geral do domínio, toda a informação relevante a respeito do domínio de aplicação do programa de mensuração é armazenada sob diversas formas. Os objetos envolvidos podem ser produtos ou processos existentes, e serão utilizados na definição dos atributos envolvidos, tanto na recuperação de experiências, como na definição de modelos de qualidade para os planos de mensuração.

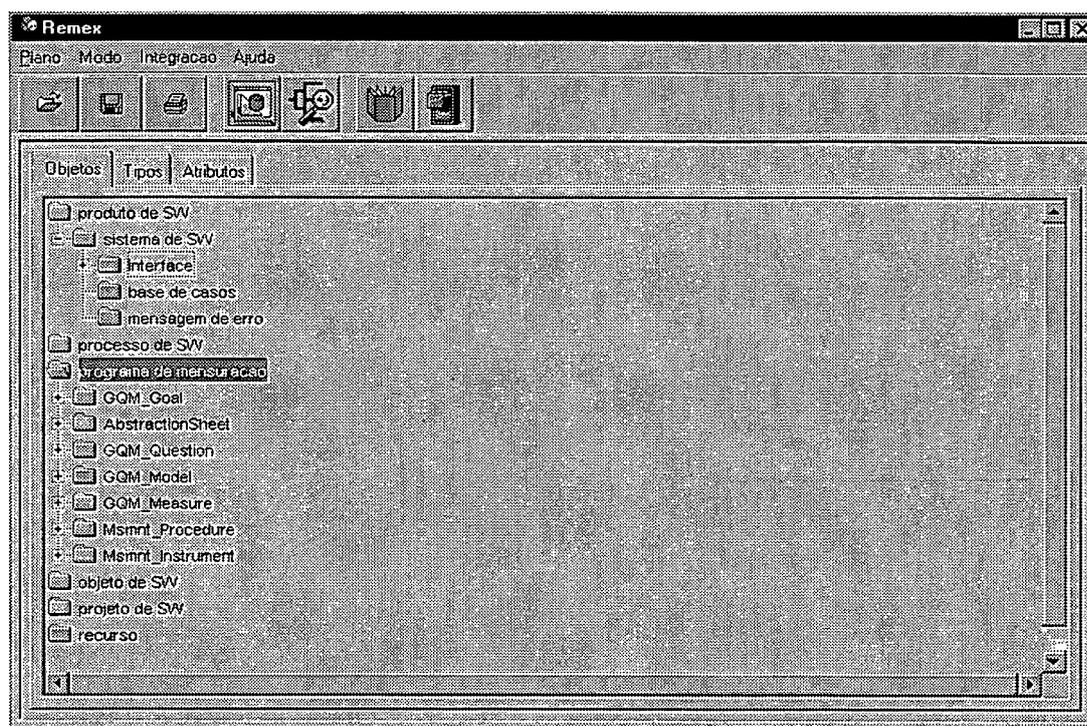


FIGURA 31. MANUTENÇÃO DO SISTEMA: HIERARQUIA DE OBJETOS

Muitos destes objetos já estão definidos na ferramenta, bastando ao engenheiro de conhecimento adicionar, os que forem necessários através da interface de manipulação de objetos apresentada na figura 31.

#### 4.2 Hierarquia de Tipos

Os tipos de informação utilizadas no sistema podem ser trabalhados no formulário de definição de tipos (ver figura 32), onde os principais tipos já estão informados, o que permite ao engenheiro de qualidade utilizar os tipos pré-definidos na definição de seus atributos, ou criar um novo tipo, com base nos tipos padrões.

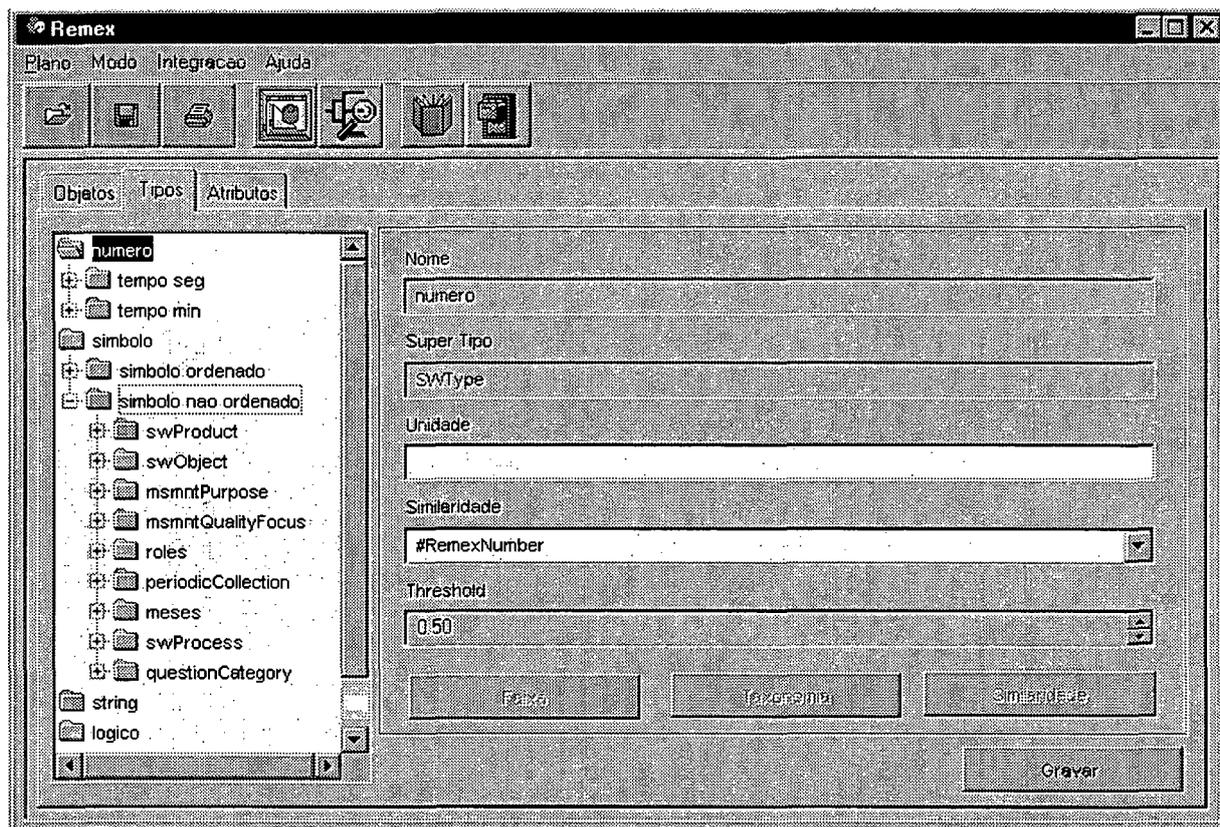


FIGURA 32. MANUTENÇÃO DO SISTEMA: DEFINIÇÃO DE TIPOS

A informação de tipo é também importante para a recuperação de informações similares, uma vez que em sua definição encontra-se a escolha do algoritmo de recuperação, e o valor mínimo (Threshold) para consideração de item similar.

#### 4.3 Atributos

No formulário de definição de atributos (ver figura 33) é realizada a manutenção dos atributos que são manipulados pelo sistema e os utilizados na definição dos modelos de qualidade e, por conseguinte nas medidas e procedimentos de mensuração.

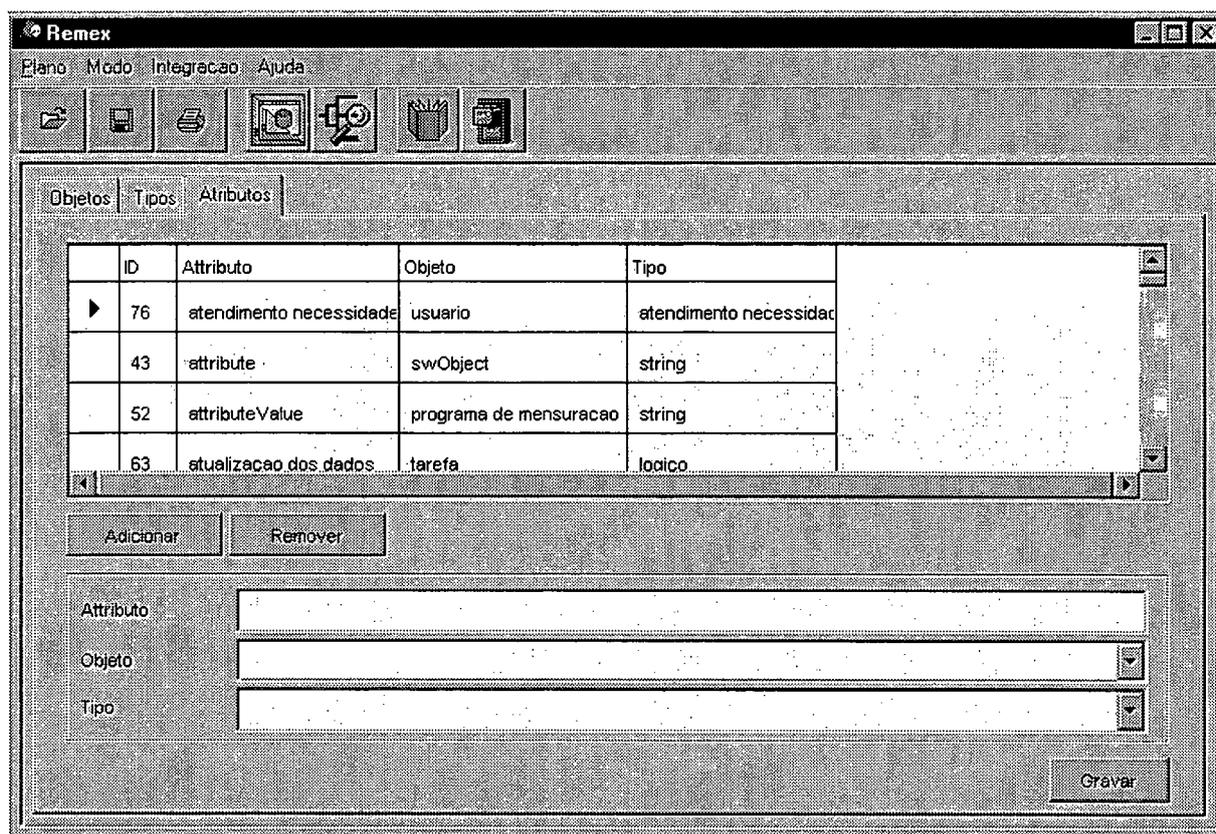


FIGURA 33. MANUTENÇÃO DO SISTEMA: DEFINIÇÃO DE ATRIBUTOS

Os atributos utilizados pelo sistema, e os mais utilizados em programas de mensuração já estão definidos no sistema, bastando ao engenheiro de conhecimento utilizar os atributos pré-definidos, ou acrescentar os que forem necessários.

#### 4.4 Metas de Recuperação

No modo base de experiências, o engenheiro de conhecimento tem a possibilidade de ajustar o funcionamento da GQM-EB. No formulário de definição das metas de recuperação (ver figura 34) é possível definir ou ajustar a forma pela qual o sistema efetuará a busca por casos similares em cada etapa do planejamento, ou prevenção/resolução de problemas.

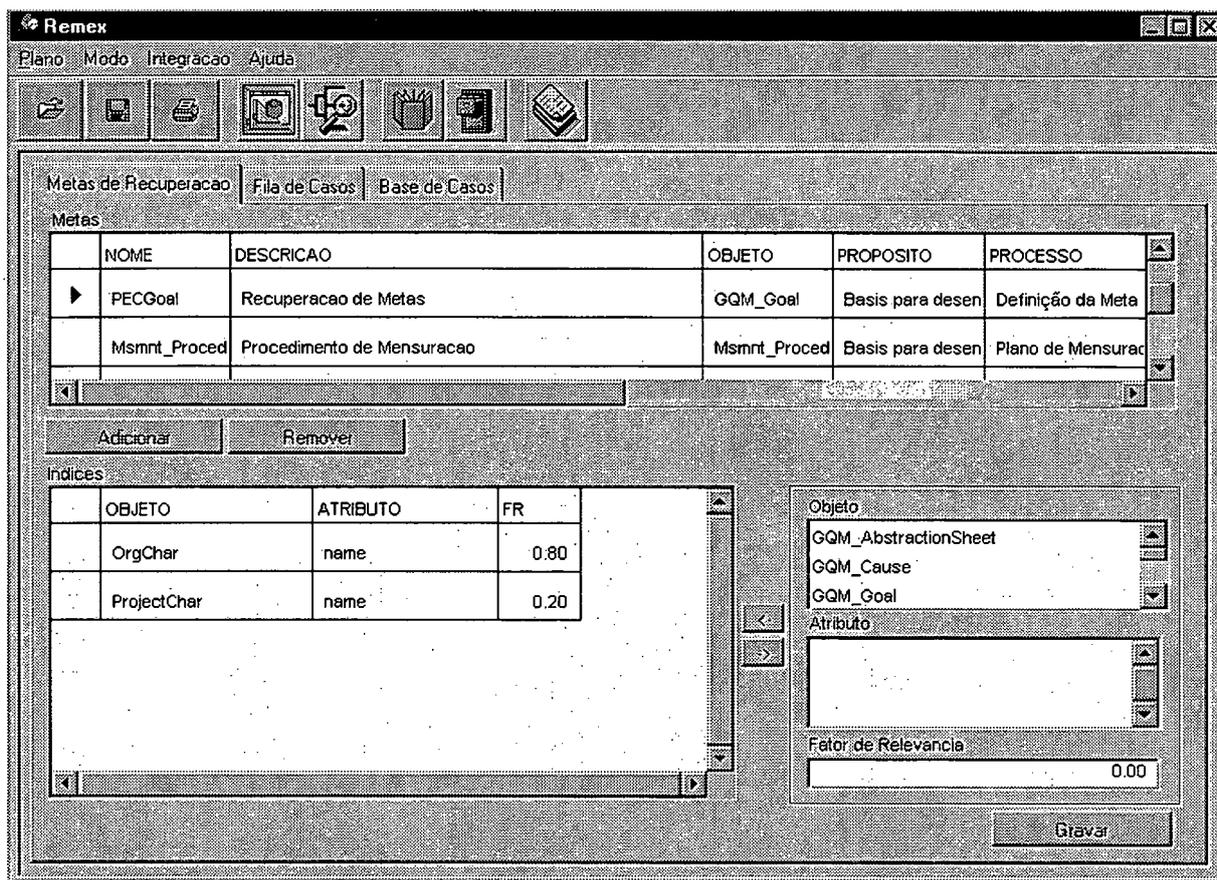


FIGURA 34. MANUTENÇÃO DO SISTEMA: DEFINIÇÃO DAS METAS DE RECUPERAÇÃO

As metas de recuperação são definidas com base nos objetos manipulados pela GQM-EB e pelo mecanismo de recuperação, tornando-o flexível e ajustável às necessidades de cada organização em particular.

#### 4.5 Base de Casos

Ainda no modo base de experiências, o engenheiro de conhecimento tem acesso ao repositório da GQM-EB (ver figura 35), para a sua devida manutenção. É possível consultar todo o conhecimento armazenado, além de que casos com pouca utilização ou relevância no contexto da organização podem ser eliminados.

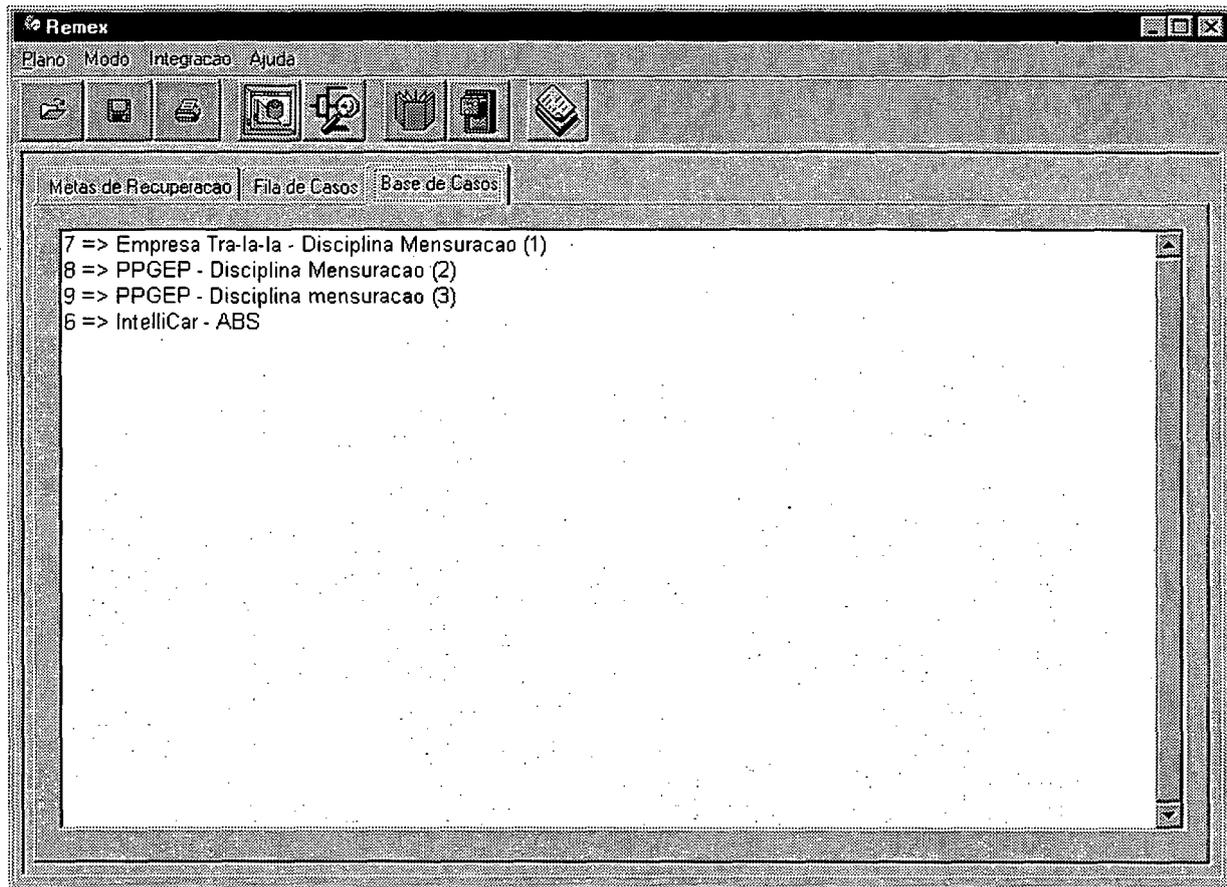


FIGURA 35. MANUTENÇÃO DO SISTEMA: BASE DE CASOS

## CONCLUSÃO

A abordagem GQM tem demonstrado ser adequada para o estabelecimento de programas de mensuração efetivos, focando a atenção nas causas reais dos problemas de processos de desenvolvimento em projetos industriais. Um considerável progresso tem sido feito no estabelecimento de mensuração na prática, porém o planejamento de programas de mensuração baseados em GQM tem sido uma tarefa complexa que consome muito tempo.

A fim de auxiliar na árdua tarefa de planejamento de programas de mensuração, foi apresentada neste trabalho uma abordagem customizável de Raciocínio Baseado em Casos para a operacionalização da reutilização sistemática e aprendizagem organizacional de experiências de mensuração incorporada ao processo de planejamento GQM.

A ferramenta desenvolvida deu suporte a todas as tarefas do processo de planejamento de mensuração, guiando a edição dos respectivos produtos GQM e possibilitando uma recuperação customizável de casos de experiências de mensuração similares, para várias metas de reutilização.

A aquisição contínua de novas experiências e sua integração à base de experiências, levando em consideração as características específicas do domínio, tal como a falta de modelos de domínios na prática, diversidade de ambientes e tarefas a serem suportadas, dados incompletos e consideração da similaridade das experiências, permitiu a implementação de uma Fábrica de Experiências que possibilita o compartilhamento do conhecimento por toda uma organização, favorecendo o aprendizado corporativo.

O comparativo efetuado entre as diversas ferramentas existentes, demonstrou que,

em sua grande maioria, estas ferramentas apresentam suporte ao ciclo de planejamento de programas de mensuração GQM, porém não oferecem a possibilidade de reutilização do conhecimento armazenado nos planos de mensuração já efetuados, bem como das lições aprendidas na resolução de problemas ocorridos, sem possibilitar a disseminação deste conhecimento através da organização.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [AK99] ABIB, Janaina C., Kirner, Tereza G. GQM-PLAN: Uma ferramenta para Apoiar Avaliações de Qualidade de Software. Departamento de Computação. Universidade de São Carlos. 1999
- [Alt97] K.-D, Althoff. Evaluating Case-Based Reasoning Systems: The Inreca Case Study. Postdoctoral Thesis (Habilitationsschrift), University of Kaiserslautern, Germany, 1997.
- [AP94] AAMODT, Agnar, Enric Plaza.(1994) :Case Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches.
- [AW91] K.-D. Althoff, S. Wess. Case-based Knowledge Acquisition, Learning and Problem Solving in Diagnostic Real World Tasks. Proc. Of the 5<sup>th</sup> European Knowledge Acquisition for Knowledge-Based Systems Workshop, Scotland/UK, 1991.
- [BC91] Caldiera and V.R. Basili. "Identifying and Qualifying Reusable Components". IEEE Software, February 1991. pp. 61-70.
- [BCR94] V. R. Basili, G. Caldiera, H. D. Rombach. Experience Factory. In John J. Marciniak (ed.), Encyclopedia of Software Engineering, vol.1. John Wiley & Sons, 1994.
- [BCR94a] R. Basili. "Quantitative Evaluation of Software Engineering Methodology, " Proceedings of the First Pan Pacific Computer Conference, Melbourne. Austrália, September, 1985.
- [BCR94b] V.R. Basili, G. Caldiera, H.H. Rombach. Goal/Question/Metric Approach. In John

- J. Marciniak, editor, Encyclopedia of Software Engineering, volume 1. John Wiley & Sons, 1994.
- [BDR96] L.C. Briand, C. M. Differing, H.D. Rombach. Practical Guidelines for Measurement-Based Process Improvement. *Software Process*, 2(4):253-280, December, 1996.
- [BDT96] A. Bröckers, C. Differing, G. Threin. The Role of Software Process Modeling in Planning Industrial Measurement Programs. *Proceedings of the METRICS'96. ICSE*, 1996.
- [Ber+97] R. Bergmann et al. Initial methodology for Building and Maintaining a CBR application. *ESPIRIT Project 22196*, 1997.
- [Boe81] Boehm, B.W.: *Software Engineering Economics*. Advances in Computing Science and Technology. Prentice Hall, 1981.
- [BW84] Basili, V.R.; Weiss, D.M.: A Methodology for Collection Valid Software Engineering Data. *IEEE Transactions on Software Engineering*, SE-10 (6), November 1984, 728-738.
- [CEM96] CEMP Consortium, Customized Establishment of Measurement Programs. Final Report, *ESSI Project Nr. 10358*, 1996.
- [CFG+98] G. Cugola, C. Gresse, P. Fusaro, <sup>a</sup> Fuggetta, L. Lavazza, S. Manca, M.R. Pagone, G. Ruhe, R. Soro. A Case Study of Evaluating Configuration Management Practices with Goal-Oriented Measurement. In *Proceedings of the 4th Int. IEEE Symposium on Software Metrics*, 1997.
- [Das92] M. K. Daskalantonakis. A practical view of software measurement and implementation experiences within Motorola. *Transactions on Software Engineering*, 18(11):998-1010, November 1992.
- [DEM86] Edwards Deming, *Out of the Crisis*. MIT Center for Advanced Engineering Study, MIT Press. Cambridge. MA, 1986.
- [Dif93] Differding C. An Object Model for Supporting the GQM Paradigm (in German). Master Thesis, Department of Computer Science, University of Kaiserslautern, 67653. Kaiserslautern, Germany, June 1993.

- [FLM+96] Fuggetta, L. Lavazza, S. Morasca, S. Cinti, G. Oldano and E. Orazi. Applying QM in an Industrial Software Factory. (submitted to IEEE Transactions on Software Engineering).
- [GAB99] C. Gresse von Wangenheim, K-D. Althoff, R. M. Barcia. Intelligent Retrieval of Software Engineering Experienceware. Technical Report PPGEP-C3001.99E. Graduate Program in Production Engineering, Federal University of Santa Catarina, Brazil, 1999.
- [GB98] Gresse, Briand Requirements for ... Elsevier..
- [Gre00] C. Gresse von Wangenheim, A. von Wangenheim. Introducing Effective Measurement-Based Effort and Schedule Management in Small Software Companies. Technical Report GeNESS001.00E, Centro GeNESS, Universidade Federal de Santa Catarina, 2000.
- [Gre98] C. Gresse von Wangenheim, Knowledge Management in Experimental Software Engineering – Create, Renew, Build and organize Knowledge Assets. Proc. Of the 10<sup>th</sup> Int. Conference on Software Engineering and Knowledge engineering, San Francisco, 1998.
- [Gre99] C. Gresse von Wangenheim. REMEX - A Case Based Approach for Reuse of Software Measurement Experienceware. Proc. of 3rd Int. Conf. on Case-Based Reasoning, Germany, 1999.
- [GRR98] C.Gresse von Wangenheim, H.D. Rombach and G. Ruhe. Tutorial em Melhoramento de Software Baseado em Mensuração – Como Aplicar GQM na Prática ?. Proceedings of the IX CITS – Conferência Internacional de Tecnologia de Software: Qualidade de Software, Curitiba, Brazil, 1998.
- [HOR+96] Hoisl, B. et Al.: No Improvement without Feedback: Experiences from Goal-Oriented Measurement at Schlumberger. Fifth European Workshop on Software Process Technology, Nancy (France), October 1996.
- [KA83] Kogure, M.; Quality Function Deployment and CWQC in Japan. Quality Progress, October 1983
- [Kol93] KOLODNER, J. (1993): Case-Based Reasoning. San Francisco: Morgan Kaufmann Publishers.

- [LSO+98] F. van Latum, R. Van Solingen, M. Oivo, B. Hoisl, H.D. Rombach, G. Ruhe. Adopting GQM-Based Measurement in an Industrial Environment. IEEE Software, vol. 15, no. 1, January/February 1998.
- [Mar95] MARIS, Marco van GQM-DIVA-Ein Werkzeug zur Definition, Interpretation und Validation von GQM-Planen. Diplomarbeit, Universität Kaiserslautern, Fachbereich Informatik. 1995.
- [NAS94] NASA, National Aeronautics and Space Administration. Software Measurement Guidebook. Technical Report SEL-84-101, NASA Goddard Space Flight Center, Greenbelt MD 20771, July 1994.
- [OB92] OIVO, Markku & BASILI, V.R. Representing Software Engineering Models: The TAME goal oriented approach. IEEE Transactions on Software Engineering, 18(10):886-898, October, 1992.
- [Oiv90] OIVO, Markku. Knowledge-based Support for Embedded Computer Software Analysis and Design. VTT Publication 68, Espoo, September. 1990. ISBN 951-38-3763-7
- [PB86] Porter, B. and Bareiss, R. (1986) PROTOS: An experiment in knowledge acquisition for heuristic classification tasks. In: Proceedings of the First International Meeting on Advances in Learning (IMAL), Les Arcs, France, pp. 159-174.
- [PER96] The PERFECT Consortium. Goal Oriented Measurement Using GQM. ESPIRIT Project no. 9090 "PERFECT", Kaiserslautern, Germany, 1996.
- [Qui86] Quinlan, J.R. (1986): Induction of Decision Trees. Machine Learning, 1(1), 81-106.
- [Ric95] M.M. Richter. On the Notion of Similarity in Case-Base Reasoning. G. Della Riccia et. al. (eds), Mathematical and Statistical Methods in Artificial Intelligence, Springer Verlag, 1995.
- [Sch82] Schank, R. (1982): Dynamic Memory: A Theory of Learning in Computers and People. Cambridge University Press.
- [SLO95] R. van Soligem, F. Van Latum, M. Oivo and E. Berghout. Application of Software

Measurement at Schlumberger RPS: Towards Enhancing GQM. Proceedings of the sixth European Software Cost Modelling Conference (ESCOM), May 1995.

[TG99] C. Tautz, C. Gresse von Wangenheim. REFSENO: A Representation Formalism of Software Engineering Ontologies. Proc. Of 5<sup>th</sup> German Conference on Knowledge-Based Systems, 1999.

[Tve77] A. Tversky. Features of Similarity. Psychological Review, 84, 1977.

[Wat97] Ian Watson (1997): Applying Case Base Reasoning: Techniques for Enterprise Systems.

[Wes95] S. Wess. Fallbasiertes Problemlösen in wissensbasierten Systemen zur Entscheidungsunterstützung und Diagnostik. Ph.D. Thesis, University of Kaiserslautern, Germany, infix Verlag, 1995.

