

# Um Modelo de Classes para um Ambiente de Geração de Programas de Medição de Software Baseados na Web

Luiz Paulo Alves Franca  
Arndt von Staa  
Hamilton José Sales Fonte II  
{franca,arndt}@inf.puc-rio.br  
hamilton@les.inf.puc-rio.br

Departamento de Informática, PUC-Rio

## Resumo

*O aumento de atenção no processo de desenvolvimento de software através da adoção de modelos de maturidade, gerou um crescimento da demanda por sistemas de apoio à qualidade de software. Neste trabalho, apresentamos um modelo de classes adequado à construção de sistemas de medição, satisfazendo os requisitos: Processo de produção flexível; Interface amigável e integrada ao ambiente de desenvolvimento; Baixo custo possibilitando a adoção por pequenas organizações de software. O modelo de classes deve suportar ainda a evolução e a integração de diferentes programas de medição construídos pelo ambiente. O modelo proposto é utilizado por COMPASSO: Um Ambiente de Elaboração de Programas de Medição Baseados na Web. Este ambiente deriva programas de medição específicos a partir de um modelo genérico instanciável. Tanto o modelo genérico como os modelos instanciados podem ser evoluídos, mesmo quando já existirem bases de medição povoadas. COMPASSO visa fornecer instrumentos para estabelecer, povoar e explorar sistemas de medição eficazes e viáveis em pequenas organizações.*

## 1. Introdução

A medição de software está se consolidando como uma prática importante para suportar as iniciativas de melhorias no processo de software realizadas por muitas empresas [1]. A medição de software, pode servir tanto como fonte de informação para o monitoramento da situação atual do processo de software, como para a identificação de desvios na execução deste processo.

Geralmente, a introdução de um programa de medição de software numa organização, é marcado por dificuldades inicialmente na definição do programa e, posteriormente, na execução do programa. Em relação a definição, muitas vezes, a empresa não consegue identificar as métricas mais relevantes para sua realidade. Para ajudar a atividade de definição, algumas propostas já estão sendo utilizadas com sucesso [2, 3, 4, 5, 6, 7]. Para a execução do programa de medição, o suporte computacional é fundamental, pois a medição de software envolve a geração e a manipulação de um grande volume de dados. Contudo, existe muita dificuldade de conseguir ferramentas adequadas para suportar o programa de medição requerido pela organização. Muitas vezes o programa de medição acaba sendo definido para adequar-se a capacidade das ferramentas selecionadas.

No sentido de auxiliar uma organização para superar os problemas na introdução do seu programa de medição, construímos um ambiente de suporte a programas de medição de software, denominado COMPASSO, que permitirá a customização do programa para realidade da em-

presa ou do projeto em estudo. Os sistemas de suporte a programas de medição instanciados a partir do ambiente, estarão preparados para realizar as atividades de coleta e análise de dados, podendo estas atividades serem realizadas localmente ou remotamente através da Web.

Neste artigo apresentaremos, o modelo de classes do COMPASSO. O modelo de classes proposto está preparado para atender as necessidades de extensão e integração dos programas de medição de software. O modelo além de tratar os aspectos gerais da medição de software, foi também preparado para atender as restrições de nosso usuário-alvo, que é uma incubadora de empresas de software.

O projeto do nosso ambiente foi baseado na ferramenta PSM [14] e nas estratégias de desenvolvimento de ferramentas de coleta de dados de engenharia de software através da Web [12,13].

Este artigo apresenta o modelo de classes do ambiente da seguinte forma: a seção 2 apresenta uma visão geral do ambiente; a seção 3 e 4 descrevem o modelo; a seção 5 apresenta uma comparação com os trabalhos relacionados. Na seção 6 é apresentada a situação atual e futura deste trabalho. As conclusões são apresentadas na seção 7.

## **2. COMPASSO: Ambiente de suporte a programas de medição**

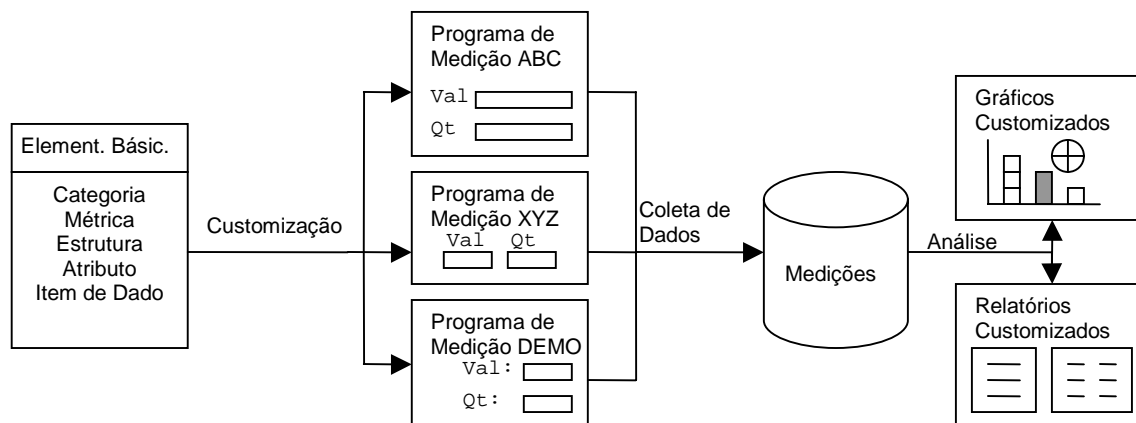
Para facilitar o entedimento das motivações para definição do modelo proposto, faremos uma breve descrição da utilização do ambiente COMPASSO.

A primeira etapa na utilização do ambiente é a definição do programa de medição. O ambiente já vem instalado com alguns programas de medição. A partir de um programa modelo, o usuário pode customizar o programa através da inclusão ou exclusão de componentes do programa. Durante o processo de definição o usuário, conforme necessário, pode acrescentar novos componentes para o programa de medição. Neste caso, a ferramenta permite ao usuário criar novos componentes (*Categoria, Estrutura, Métrica, Atributo, Item de dado e Indicador*). Nesta etapa, as regras de validação das medições e também a frequência destas medições são definidas.

Antes de iniciar a medição de software de um dado projeto, a Organização tem que selecionar o programa de medição mais adequado para o acompanhamento do projeto. Após a seleção do programa de medição, o ambiente gera uma instância de sistema de medição contendo apenas os elementos presentes no programa de medição selecionado. Por exemplo, o programa de medição selecionado está focado somente nas métricas relacionadas com esforço ( ex: homem/hora), neste caso, a instância do sistema de medição somente irá apresentar telas para aquisição de dados relacionados com esforço, todo o conjunto restante de métricas não aparece nesta versão customizada. Na medida que a medição de software vai amadurecendo dentro da organização, o programa de medição vai sendo ampliado, através do acompanhamento de mais métricas[11].

Utilizando o sistema de medição customizado, o usuário preenche os formulários de medição em função da periodicidade definida. Esses formulários alimentam a base de dados de medição. Através dos indicadores de medição (ex.: gráficos, planilhas) definidos no programa de medição, o usuário pode analisar a evolução do projeto e de seus produtos.

A figura 1 mostra uma visão geral do ambiente COMPASSO, no qual podem ser configurados diferentes programas de medição através da customização de elementos básicos. A coleta de dados realizadas através desses programas registram as medições num único banco de dados, possibilitando a extração de estatísticas integradas



**Figura 1. Visão Geral do COMPASSO Envi-**

COMPASSO disponibiliza a base de medições através de um servidor Web. Por intermédio de browsers, COMPASSO está preparado para 2 modos de operação:

- **Modo Internet/Intranet:** O usuário está conectado ao servidor Web da organização, acessando e alterando a base central de medições.
- **Modo Local:** O usuário acessa e altera a base de medições local. Através de um mecanismo de *upload*, as alterações na base local são transferidas para base central do servidor Web da organização.

Os 3 principais módulos do COMPASSO são: Customizador, Medidor e Apresentador.

- **Customizador:** realiza as atividades de definição e customização dos programas de medição. Este módulo somente pode ser operado no modo Internet/Intranet;
- **Medidor:** realiza a coleta e a análises das medições. O módulo Medidor varia em função do programa de medição selecionado. Este módulo pode ser operado tanto no modo Internet/Intranet como no modo Local. A atividade de medição de software, pode ser feita de 2 formas: tela tipo formulário ou por importação de arquivo texto de medições;
- **Apresentador:** Seleciona, agrega e exhibe os dados de medição em diversos formatos. Este módulo somente pode ser operado no modo Internet/Intranet.

Em função das características de nosso usuário-alvo: uma incubadora de empresas de software [9,10], tivemos que preparar o modelo de dados para ser multiempresa, pois é esperado que as empresas incubadas adotem a medição de software durante o período de incubação e, posteriormente, quando se estabelecerem no mercado. Estas pequenas organizações devem adotar programas de medição de forma evolutiva [8, 11]. Inicialmente realizarão as medições mais elementares e, à medida que forem adquirindo maturidade, passarão a programas de medição cada vez mais complexos. Por outro lado, a base de dados da incubadora, estará preparada para armazenar as medições tanto das empresas incubadas como da ex-incubadas. Desta forma, a base de dados poderá ser utilizada para o desenvolvimento de modelos de engenharia de software aplicada a pequenas organizações, operando em vários níveis de abstração.

### 3. Estrutura básica do Modelo de Classes

Para atender aos requisitos básicos da ferramenta [9,10], tivemos que fazer um modelo de classes que garantisse a evolução e a integração das métricas coletadas. O modelo proposto adota a abordagem da ferramenta PSM [14] que caracteriza uma métrica como sendo uma

composição (Tabela 1 ) formada por: *Estruturas*, *Atributos*, e *Itens de Dado*. Dentro do nosso trabalho, estes componentes apresentam as seguintes definições:

- Estruturas: São as entidades do processo de software às quais estão sendo aplicadas métricas. Uma estrutura pode corresponder a um artefato ou à uma fase (atividade) dentro do projeto;
- Atributos: São características contextuais de uma medição da métrica;
- Itens de dado: São os valores da medição realizada segundo uma métrica.

ESFORÇO		TAMANHO DE PRODUTO	
Estrutura	Atividade	Estrutura	Sistema
Estrutura	Sistema	Atributo	Versão
Atributo	Organização	Atributo	Linguagem
Atributo	Tipo de Trabalho	Item de dado	Total de linhas de código
Item de dado	Total de horas trabalhadas	Item de dado	Total de linhas de código adicionadas

**Tabela 1.** Exemplo de composição das métricas Esforço e Tamanho de Produto

Através do ambiente será permitida a definição de diferentes métricas e a criação de diversos programas de medição. Portanto, a composição de uma métrica pode ser redefinida para cada programa de medição. Na tabela 2 é exemplificada a redefinição da métrica *Tamanho de Produto*, na qual o programa de medição ABC retira um atributo e acrescenta dois itens de dados em relação ao programa XYZ.

PROGRAMA DE MEDIÇÃO XYZ		PROGRAMA DE MEDIÇÃO ABC	
TAMANHO DO PRODUTO		TAMANHO DO PRODUTO	
Estrutura	Sistema	Estrutura	Sistema
Atributo	Linguagem	Atributo	Linguagem
Atributo	Versão	Item de Dado	Total de linhas de código novas
Item de Dado	Total de linhas de código	Item de Dado	Total de linhas de código alteradas
Item de Dado	Total de linhas de código Novas	Item de Dado	Total de linhas de código excluídas

**Tabela 2.** Exemplo da customização da métrica do tamanho do produto

#### 4. COMPASSO: Modelo de Classes

O modelo de classes foi elaborado considerando as necessidades de evolução e expansão resultantes da customização de um programa de medição. Adotamos a notação da UML[18] para descrever os modelos. Outro aspecto considerado foi a necessidade de garantir a organização e a estruturação das atividades de coleta e análise das medições realizadas. Para facilitar a descrição, dividimos o modelo em 4 grupos:

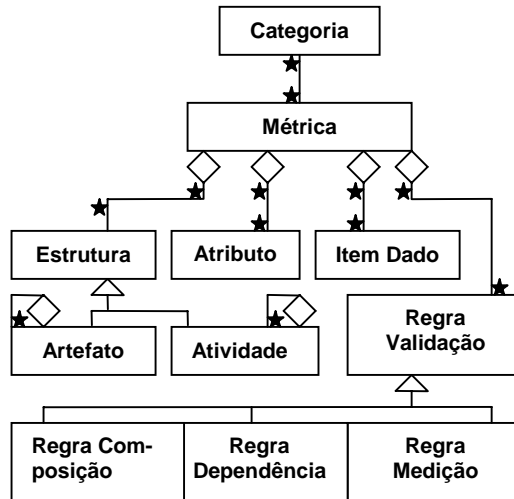
- Elementos Básicos;
- Programa de Medição;
- Medição;
- Administrativo

##### 4.1 Grupo Elementos Básicos

O grupo *Elementos Básicos* engloba as classes utilizadas na customização de um programa de medição. Neste grupo são registrados os componentes de uma métrica. Este grupo corresponde ao conjunto de componentes que é a base comum para geração de diferentes programas de

medição. Novas métricas que forem surgindo podem ser acrescentadas ao conjunto inicial de elementos básicos. A figura 2 apresenta o Diagrama de Classes deste grupo.

Neste grupo foi dada ênfase à questão das regras de validação, pois elas são ferramentas importantes para garantia da qualidade da base de medição. Segue abaixo a descrição de cada classe deste grupo.



**Figura 2.** Diagrama de Classes do Grupo Elementos Básicos

#### Artefato

Classe das estruturas do tipo Artefato que estão sujeitos à medição. Um artefato pode ser tanto um produto final (ex: sistema, programa, etc.) como um produto intermediário (ex: especificação, plano de teste, etc) do processo de desenvolvimento. A divisão de um sistema em módulos e em funções é um exemplo de estrutura do tipo Artefato.

#### Atividade

Classe das estruturas do tipo Atividade relevantes para a medição. A divisão de um sistema em fases e em atividades é um exemplo de estrutura do tipo Atividade.

#### Atributo

Classe de características de métricas e que distingue uma medição. Exemplos: Estimado/Real, versão, linguagem.

#### Categoria

Classe dos conjuntos de métricas relacionados com um determinado conceito. Exemplos: Esforço, Tamanho. Dentro da categoria *Esforço* estão todas as métricas relacionadas com este conceito.

#### Estrutura

Classe genérica das formas de organização de um projeto. Uma estrutura pode representar, por exemplo, as atividades e as sub-atividades do projeto, a divisão do sistema em módulos e em programas. Uma medição está sempre relacionada a algum nível de uma estrutura.

#### Item de Dado

Classe dos itens quantificados da métrica cujos valores são obtidos através de medições. Por exemplo: na métrica *Tamanho de Programa*, o *Total de Linhas* é um Item de dado desta métrica. Na métrica *Esforço*, o *Número de Horas Trabalhadas*, e o *Número de Horas Extras* são exemplos de itens de dado.

### Métrica

Classe características de projeto ou de produto que podem ser monitoradas. Os componentes de uma métrica são: *Estruturas*, *Atributos*, e *Itens de dado*. A composição da métrica depende do programa de medição.

### Regra de Validação

Classe genérica das regras de validação a serem aplicadas a uma métrica. Exemplo: total de linhas de código *é um* valor numérico.

### Regra de Composição

Classe das regras de validação referentes aos elementos que compõem uma métrica. Exemplo: valores máximo e mínimo de um determinado item de dado.

### Regra de Dependência

Classe das regras de validação envolvendo dois ou mais elementos que compõem uma dada métrica. Exemplo: na métrica *Tamanho do Produto* (Tabela 1), caso o atributo *Linguagem* seja igual a JAVA, os valores máximo e mínimo do item de dado *Total de Linhas* serão 20 e 500 respectivamente.

### Regra de Medição

Classe das regras de validação envolvendo medições já realizadas. Exemplo: na métrica *Fases Completadas*, caso o *final da fase de análise* tenha sido medido em dezembro de 1998, o valor medido *final da fase de programação* deverá ser posterior a dezembro de 1998.

## **4.2 Grupo Programa de Medição**

O grupo Programa de Medição ( Figura 3) engloba as classes correspondentes aos diversos programas de medição que uma organização pode criar através do ambiente COMPASSO. Diversos programas de medição podem ser customizados a partir da combinação dos elementos básicos ( seção 4.1). Um programa de medição define a forma na qual as medições vão ser coletadas e analisadas. Uma mesma organização pode apresentar diferentes programas de medição em função do tipo de projeto que está sendo monitorado.

Na Figura 3 as classes que estão definidas em outro grupo apresentam a identificação da seção no topo direito da classe. Fugimos ligeiramente da notação UML para facilitar a legibilidade do diagrama. Neste grupo, a composição de uma métrica pode ser redefinida no sentido de adequar-se aos objetivos do programa de medição que está sendo elaborado.

### Atributo do Programa de Medição

Classe dos Atributos da Métrica para um programa de medição específico.

### Categoria do Programa de Medição

Classe das Categorias para um programa de medição específico.

### Estrutura do Programa de Medição

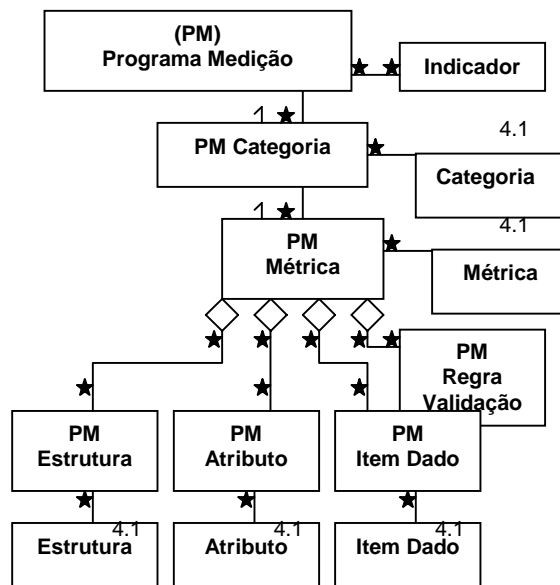
Classe que representa as Estruturas da métrica para um programa de medição específico.

### Item de Dado do Programa de Medição

Classe dos Itens de Dado da métrica de um programa de medição específico. Esta Classe contém as regras de apresentação e de validação do item de dado para um determinado programa de medição.

### Indicador de Medição

Classe dos Gráficos e/ou Relatórios específicos que serão gerados a partir das medições contidas na base de medição.



**Figura 3.** Diagrama de Classes do Grupo Programa de Medição

#### Métrica do Programa de Medição

Classe das Métricas de um programa de medição específico.

#### Programa de Medição

Classe do conjunto de processos, métodos, técnicas, e ferramentas empregados na medição de software

#### Regra de Validação do Programa de Medição

Classe genérica das regras de validação de um programa de medição específico.

#### Regra de Composição do Programa de Medição

Classe das regras de validação referentes aos elementos que compõem uma métrica de um programa de medição específico.

#### Regra de Dependência do Programa de Medição

Classe das regras de validação cuja aplicação envolve dois ou mais itens de uma métrica de um programa de medição específico.

#### Regra de Medição do Programa de Medição

Classe genérica das regras de validação envolvendo medições já realizadas segundo um programa de medição específico.

### **4.3 Grupo Medição**

Um dos resultados da adoção de um programa de medição, é a criação de uma base de dados contendo o histórico dos projetos medidos. O grupo Medição engloba as classes que estão envolvidas no registro das medições realizadas. Esta parte do modelo foi elaborado considerando dois aspectos:

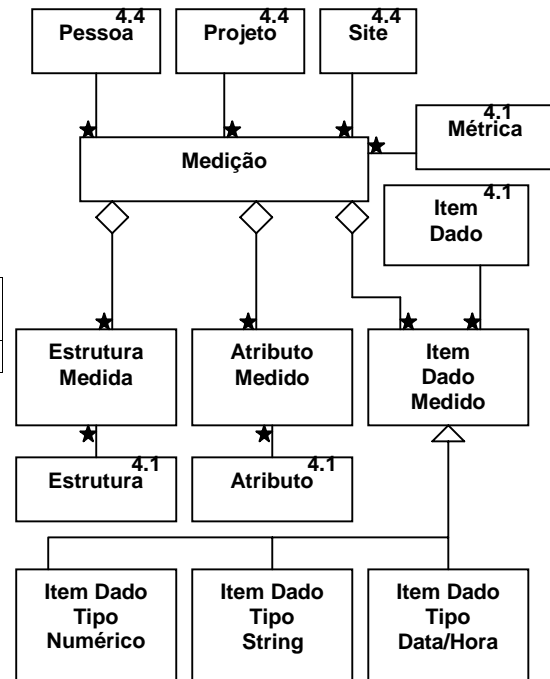
- facilidade de alteração do programa de medição após ele já ter sido iniciado;
- possibilidade de integração das medições realizadas através de diferentes programas de medição.

No ambiente COMPASSO, a base de medições é comum para todos os programas de medição, isto é, todos os diferentes programas de medição de uma organização alimentam uma mesma base de dados. A tabela 3 apresenta um exemplo de registros do programa de medição XYZ (Tabela 2) para a métrica Tamanho do Produto.

Na figura 4, a classe Medição reflete a composição da métrica que está sendo medida. Com vistas a facilitar o tratamento da classe Item de Dado Medido, esta classe foi especializada levando a várias classes correspondentes a tipos de dados primitivos. Esta especialização, além de servir como ponto de expansão do modelo, permite também uma otimização nas operações de análise das medições, que poderão utilizar operações próprias para cada tipo de dado primitivo.

Data	Sistema	Ver	Ling	Total de LC	Total de LC novas
06/98	Compra	2.0	JAVA	7000	1000

**Tabela 3.** Medições da Métrica Tamanho de Produto



**Figura 4.** Diagrama de Classes do Grupo Medição

#### Atributo Medido

Classe dos valores medidos de um atributo da métrica.

#### Estrutura Medida

Classe dos valores de um nível da estrutura na qual a métrica foi medida.

#### Item de Dado do Tipo Data

Classe dos valores medidos do tipo Data.

#### Item de Dado do Tipo Numérico

Classe dos valores medidos do tipo Numérico.

#### Item de Dado do Tipo String

Classe dos valores medidos do tipo String.

#### Medição

Classe que representa a medição (coleta) de uma dada métrica.



Para exemplificar, as tabelas 4,5,6,7, ilustram como os registros das medições da tabela 3 poderiam estar distribuídos por estas classes.

ID Medição	Data	Métrica	Projeto	Site	Pessoa
0001	06/98	Tamanho Produto	Cont	C	Paulo
0002	06/98	Tamanho Produto	Cont	C	Rose

Tabela 4. Objetos da Classe Medição

ID Medição	Sistema
0001	Compra
0002	Venda

Tabela 5. Objetos da Classe Estrutura Medida

ID Medição	Atributo	Valor
0001	Versão	2.0
0001	Linguagem	JAVA
0002	Versão	3.0
0002	Linguagem	C++

Tabela 6. Objetos da Classe Atributo Medido

ID Medição	Item Dado	Valor
0001	Total de LC	7000
0001	Total de novas LC	1000
0002	Total de LC	5000
0002	Total de novas LC	750

Tabela 7. Objetos da Classe Item de Dado Tipo Numérico

```

SELECT M.IDMEDICAO, E001.VLNIVEL, A001.VLATRMED, A002.VLATRMED,
       N001.VLIITEMNUM, N002.VLIITEMNUM
FROM MEDICAO M, ESTRUT_MEDICAO E001,
     ATR_MEDICAO A001, ATR_MEDICAO A002,
     NUMDATA_MEDICAO N001, NUMDATA_MEDICAO N002
WHERE M.IDMEDICAO = E001.IDMEDICAO AND M.IDMEDICAO = A001.IDMEDICAO AND
      M.IDMEDICAO = A002.IDMEDICAO AND M.IDMEDICAO=N001.IDMEDICAO AND
      M.IDMEDICAO = N002.IDMEDICAO AND E001.IDLEVEL = 3 AND
      A001.IDATTRIB = 5 AND A002.IDATTRIB = 4 AND
      N001.IDITEM = 3 AND N002.IDITEM = 6

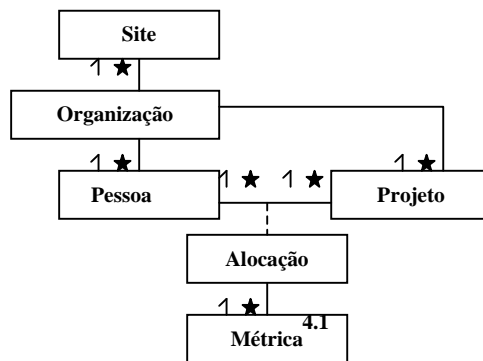
```

Figura 5. Comando SELECT para gerar a Tabela 3

O comando SELECT para listar as medições (Figura 5) é uma *query* complexa envolvendo vários JOINS com as diferentes tabelas que compõem a medição. Este comando, apesar de ser complexo, apresenta uma estrutura padrão que pode ser montada automaticamente a partir do modelo de classes. No exemplo acima, as tabelas que compõem a cláusula FROM são as tabelas correspondentes aos componentes da métrica, cada tabela recebe um alias padronizado: *Ennn* para Estrutura, *Annn* para Atributo, e *Innn* para Item de Dado. A cláusula WHERE pode ser dividida em 2 partes: A primeira corresponde aos JOINS entre a tabela medição e cada tabela componentes, e a segunda parte corresponde ao filtro específico para cada componente.

#### 4.4 Grupo Administrativo

Para garantir a qualidade da base de medição é importante que as atividades de medição possam ser monitoradas e controladas. O grupo Administrativo (Figura 6) engloba as classes que estão envolvidas no processo de coleta e análise das medições. Neste grupo estão registradas as informações contextuais da execução da medição, como por exemplo, executor da medição, responsável pela medição, local da medição.



**Figura 6.** Diagrama de Classes do Grupo de Elementos Administrativo

### Alocação

Classe das as pessoas alocadas em um Projeto. Uma pessoa pode estar alocada em mais de um projeto. Durante a medição de um projeto, uma métrica pode ser de responsabilidade de uma ou mais pessoas.

### Organização

Classe das organizações que adotam um programa de medição.

### Pessoa

Classe das pessoas participantes de um programa de medição.

### Projeto

Classe dos projetos de software da organização que serão acompanhados.

### Site

Classe dos locais onde as medições são realizadas. Uma mesma pessoa pode trabalhar em projetos que estão distribuídos em diferentes *sites*, por exemplo: laboratório de pesquisa, parceiro industrial, e casa.

## 5. Trabalhos Relacionados

Em relação à busca de trabalhos correlatos, concentramos nossos estudos em ferramentas de medição que fossem de baixo custo e de fácil instalação e que poderiam ser utilizadas pelo nosso cliente-alvo, a incubadora de empresas. Dentro deste escopo analisamos as ferramentas descritas a seguir.

### 5.1 PSM: Practical Software Measurement [14]

Esta ferramenta foi a base de nosso trabalho. A sua principal característica é a possibilidade criação de programas de medição customizados para cada projeto em estudo. A ferramenta possui *Wizards* tanto para a customização do programa de medição como para a criação de gráficos. A versão atual da ferramenta é Windows-based. As principais diferenças entre a estrutura de dados da ferramenta PSM e nossa proposta são:

- Em PSM a tabela de Medições é separada por projeto. Na nossa proposta, as medições não estão separadas por projeto;
- Em PSM as medições são armazenadas em tabelas específicas por métrica;

- Em PSM a tabela das medições de uma dada métrica tem colunas que correspondem aos atributos e itens de dados que compõem a métrica. Na nossa proposta, uma métrica está particionada em tabelas uma para cada elemento componente;
- Em PSM o relacionamento Medição e Estrutura é de 1:1. Na nossa proposta é de 1:N.
- Os aspectos administrativos da medição não são tratados por PSM;
- Para cada novo projeto a ser medido, o usuário de PSM tem que configurar todo o programa de medição. Na nossa proposta podem ser criados e instanciados programas de medição padronizados.
- PSM não permite que um mesmo projeto tenha bases de medições distribuídas. Na nossa proposta, o modelo suporta a integração de bases de medição remotas.

## **5.2 PAMPA: Project Attribute Monitoring and Prediction Associate [15]**

A principal característica de PAMPA é a possibilidade de visualização dos atributos de um projeto de forma hierárquica. A ferramenta focaliza métricas de tamanho de produto. A ferramenta não permite a customização das métricas. A flexibilidade da ferramenta está no seu gerador de gráficos codificado em macros Excel. A ferramenta contém um analisador de código para programas escritos em C, os resultados deste analisador são registrados automaticamente no banco de dados da ferramenta. A versão atual da ferramenta é Windows-Based. As principais diferenças entre a estrutura de dados da ferramenta PAMPA e nossa proposta são:

- Em PAMPA a tabela de Medições é por projeto. Na nossa proposta, as medições não estão separadas por projeto;
- Em PAMPA a tabela de Medições tem colunas que correspondem aos atributos e itens de dados que compõem a métrica. Na nossa proposta, uma medição é particionada nas tabelas de seus elementos componentes;
- Em PAMPA a estrutura medida é fixa, a nossa proposta não apresenta esta limitação.
- PAMPA não permite que um mesmo projeto tenha bases de medições distribuídas. Na nossa proposta, o modelo suporta a integração de bases remotas de medição.

## **5.3 SMMIS: Software Metrics Management Information System [16]**

O SMMIS é uma ferramenta desenvolvida pelo exército americano para padronização da medição de software dentro da instituição. Através da ferramenta, os diversos centros de desenvolvimento fazem suas medições e enviam estas medições para centros de desenvolvimento hierarquicamente superiores, permitindo desta forma a geração de uma visão global das medições dentro da instituição. A ferramenta não permite a customização das métricas. A versão atual da ferramenta é DOS-Based. As principais diferenças entre a estrutura de dados da ferramenta SMMIS e nossa proposta são:

- No SMMIS as medições de um projeto estão particionadas em tabelas para cada métrica que está sendo medida. O SMMIS apresenta 17 tabelas de métricas. Na nossa proposta, uma métrica está particionada em tabelas de seus elementos componentes;
- No SMMIS, a estrutura medida é fixa, a nossa proposta não apresenta esta limitação.

## **6. Trabalho Atual e Futuro**

Estamos concluindo a 3<sup>a</sup> versão do ambiente COMPASSO. A 1<sup>a</sup> versão (1997) foi mono-usuária e “windows-based”, codificada em Visual Basic e o banco de dados em Access. A 2<sup>a</sup>

versão (1998) já foi “web-based”, codificada em HTML, JavaScript, e CGI-LUA [3]; o banco de dados em Access. A 3ª versão que estamos concluindo é completamente “web-based”, codificada em JAVA e JDBC, o banco de dados em Access ou SQLServer.

Em relação a parte visual da ferramenta, estamos utilizando dois “free-applets”: para a visualização do programa de medição na forma de árvore de diretório e para exibição de diversos tipos de gráficos. Para automatizar a coleta de dados, estamos implementando *drivers*, por exemplo para o MS-Project (medições de esforço e de custo) e semelhante à ferramenta PAMPA [15] (medições de tamanho).

Em relação a definição das regras de validação e dos indicadores de medição, pretendemos criar uma linguagem de definição de alto nível que possa ser utilizada pelo usuário final. Na versão atual, tanto a regra como o indicador são definidos através de trechos de código em JAVA. No futuro, o interpretador da linguagem de definição vai gerar o trecho de código em JAVA correspondente à regra ou ao indicador.

Os sistemas de medições customizados gerados a partir do COMPASSO serão instalados nas empresas incubadas na incubadora InfoGene da PUC-Rio. Esta instalação permitirá avaliar a flexibilidade da abordagem proposta, o grau de utilização do sistema de medição, a aplicabilidade da solução no contexto de uma pequena empresa e a interferência do sistema de medição no dia à dia de cada empresa.

## 7. Conclusões

Neste trabalho, procuramos apresentar um modelo de classes adequado para construção de sistemas de medição caracterizados por:

- Processo de produção flexível;
- Interface amigável e integrado ao ambiente de desenvolvimento;
- Baixo custo viabilizando a adoção por pequenas organizações de software.

Em relação ao modelo de classes proposto, podemos destacar os seguintes benefícios:

- Permite a evolução da base de medição através da criação de novas métricas, da redefinição da composição de uma métrica, e da definição de novos tipos de itens de dado.
- Facilita a integração das bases de medição geradas por diferentes programas de medição customizados pelo ambiente COMPASSO. Na realidade os diferentes programas são visões de uma base de medições única.
- Suporta a geração automática de ferramentas para a manipulação das bases de medição, tais como: formulários de data-entry; drivers de importação; drivers de exportação; comandos SQL.

Um dos problemas que observamos no nosso trabalho foi o das queries de consulta da base de medições serem JOINS complexos, envolvendo as diferentes tabelas que compõem uma medição. Embora a geração dessas queries seja automática a partir do modelo de classes, na medida que a base de medição cresça muito, este tipo de consulta pode apresentar problemas de desempenho. Neste caso, quando a base de medição já apresentar uma maior estabilidade, i.e., redefinição de métricas ocorrem raras vezes, pode-se otimizar o acesso às medições, através da unificação das diferentes tabelas que compõem as métricas.

## 8. Referências

- [1] Pfleeger, S.L., Jeffery, R., Curtis, B., and Kitchenham, B.; “Status Report on Software Measurement”, *IEEE Software* 14(2), 1997, pp. 33-43.

- [2] Basili, V.R. and Rombach, H.D.; “The TAME project: Towards improvement-oriented software environments”, *IEEE Transactions on Software Engineering* 14(6), 1988, pp. 758-773.
- [3] Offen, R.J. and Jeffery, R.; “Establishing Software Measurement Programs”, *IEEE Software* 14(2), 1997, pp. 45-53.
- [4] Gresse, C.; Ruhe, G.; “Análise de Custo e Benefício de Mensuração Baseada em GQM”, Anais da X Conferência Internacional de Tecnologia de Software: Qualidade de Software, 1999, pp. 109-122.
- [5] Gresse, C., Rombach, D., and Ruhe, G.; “A Practical Approach for Building GQM-Based Measurement Programs - Lessons Learned from Three Industrial Case Studies”, Tutorial do X Simpósio Brasileiro de Engenharia de Software, 1996.
- [6] Fuggetta, A., Lavazza, L., and Morasca, S.; “Applying GQM in an Industrial Software Factory”, *ACM Transactions on Software Engineering and Methodology*, 7(4), 1998, pp. 411-448.
- [7] Latum, F., Solingen, R., Oivo, M., Hoisl, B., Rombach, D., and Ruhe, G.; “Adopting GQM-Based Measurement in an Industrial Environment”, *IEEE Software*, 14(1), 1998, pp. 78-86.
- [8] Brodman, J.G. and Johnson, D.L.; “What Small Business and Organizations Say About the CMM”, *Proceedings of ICSE-16*, 1994, pp. 331-340.
- [9] Franca, L.P.A. and, Staa, A. “Software Measurement for Small Organizations”, CAiSE98 Doctoral Consortium, Pisa, 1998.
- [10] Franca, L.P.A., Staa, A., Lucena, C.J.P., “Medição de Software para Pequenas Empresas: Uma Solução Baseada na Web”, Anais do XII Simpósio Brasileiro de Engenharia de Software, Ed. SBC, 1998, pp.71-84.
- [11] Selby, R.W, Goldstein, A., and Hart, H; *The Amadeus Measurement System: STARS's Automated, Integrated Approach to Quality, Management, and Process Measurement*, April 1993.
- [12] Laitenberger, O. and Dreyer, H.; *Automated Software Engineering Data Collection Activities via the World Wide Web: A Tool Development Strategy applied in the Area of Software Inspection*, ISERN technical report, 98-12, 1998. [Http://www.iese.fhg.de/ISERN/pub/isern.biblio.html](http://www.iese.fhg.de/ISERN/pub/isern.biblio.html)
- [13] Schwarting, D.C and Guy, E.; *Guidelines for WWW-based Metrics Automation*, Informal Technical Data, STARS-PV03-A033/001/00
- [14] *Practical Software Measurement* [Http://www.psmc.com](http://www.psmc.com)
- [15] Simmons, D., Ellis, N., Fujihara, H., and Kuo, W.; *A software Measurement: A Visualization Toolkit for Project Control and Process Improvement*, Prentice Hall, New Jersey, 1997.
- [16] *Software Metrics Management Information System (SMMIS) Homepage*, [Http://www.army.mil/swmetrics/smmis.htm](http://www.army.mil/swmetrics/smmis.htm)
- [17] Eriksson, H., and Penker, M.; *UML Toolkit*, John Wiley & Sons, New York, 1998.