

APSEE-Metrics: um modelo para mensuração em processos de software

Paulo Roberto Garcia Júnior^{1,2}
Daltró José Nunes¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)

²Centro Universitário Ritter dos Reis (UNIRITTER)

paulo.rgj@gmail.com; daltro@inf.ufrgs.br

Resumo. Este trabalho apresenta um modelo para mensuração em processos de software, no contexto do projeto APSEE-Prosoft¹. O objetivo do modelo proposto é prover suporte a realização de mensuração em processos de software, através do desenvolvimento e implantação de programas de mensuração. Neste trabalho são combinados aspectos relacionados a importantes áreas da Engenharia de Software, como mensuração, processos de software, ambientes de engenharia de software centrados em processos (PSEE), bem como a utilização de métodos formais para a definição e especificação de um modelo de mensuração integrado a um ambiente de desenvolvimento de software orientado a processos (PSEE).

Palavras-chave: Mensuração de *software*, métricas, ambientes de engenharia de *software* centrados em processos.

APSEE-Metrics: a model for measurement on software processes

Abstract. This paper presents a model for software process measurement, on the context of APSEE-Prosoft project. The model's objective is providing support to measurement in software processes through the development and implant of measurement programs. This work combine aspects related to important areas of Software Engineering, as measurement, software process, process-centered engineering environment (PSEE), besides the use of formal methods to definition and specification of a measurement model integrated to a PSEE.

Keywords: Software measurement, metrics, process-centered engineering environment.

(Received August 25, 2006 / Accepted May 30, 2007)

1. Introdução

Aumentar a qualidade, desempenho e produtividade são objetivos de qualquer organização que desenvolve *software* [1], e o uso de métricas está fortemente ligado a isto. A mensuração está se consolidando como uma prática importante para suportar melhorias neste processo, servindo tanto como fonte de informação para o monitoramento do processo de *software*, como para a identificação de desvios em sua execução [2]. Métricas de diferentes aspectos do desenvolvimento podem ajudar a determinar o progresso do projeto, e podem ser usadas no gerenciamento para prover base para tomada de decisão. As métricas corretas, usadas da maneira certa, são essenciais para o sucesso de um projeto [3].

Neste sentido, inúmeras pesquisas têm sido realizadas nesta área, como por exemplo, [4] [5] [6] [8] [9] [10], entre outras, as quais apontam o uso de métricas como fundamental para a avaliação de diferentes aspectos relativos às fases do ciclo de vida do *software*, visando à otimização de esforços e custos para manter a qualidade do produto e a competitividade no mercado. Mas a utilização de métricas por si só não apresenta resultados significativos sem a criação de um programa de mensuração, ou seja, uma seqüência de ações coordenadas para realização de medições, muitas vezes baseado nos objetivos da organização, e cuidadosamente planejados, implementados e regularmente avaliado para ser efetivo [3].

¹ Projeto desenvolvido em parceria entre o Instituto de Informática da UFRGS com o Institut für Informatik - Universität Stuttgart (Alemanha).

Quando uma organização opta por iniciar um programa de mensuração, várias providências devem ser tomadas: as medidas a serem coletadas devem ser selecionadas, procedimentos de medição devem ser padronizados e documentados, a forma de armazenamento das medidas em uma base histórica precisa ser definida, e diretrizes para a análise desses dados devem ser estabelecidas [6]. Assim, uma das principais dificuldades nesta área, refere-se justamente na condução, de forma organizada e objetiva, de um programa de mensuração, principalmente visando à objetividade na obtenção de dados e informações relevantes para tomada de decisão. Neste sentido, este trabalho visa abordar estudos referentes à esta área, tendo como principal objetivo a definição de um modelo para mensuração em processos de *software*, sendo parte de uma atividade de pesquisa do Projeto PROSOFT, referente à tecnologia de processos de *software* estudadas no desenvolvimento do ambiente de automação e gerência de processos APSEE (apresentado em maiores detalhes na seção 3.1). O projeto envolve a construção de um ambiente integrado que vem servindo de laboratório para a experimentação de tecnologias que apoiam o desenvolvimento de software com alta qualidade e produtividade. A utilização de métodos formais é enfatizada na construção das ferramentas do ambiente a partir de um paradigma próprio, criado no projeto. Sob este paradigma, ferramentas CASE foram desenvolvidas para apoiar a utilização dos métodos formais e semi-formais para a construção de software, sendo então integradas ao ambiente [11].

Este artigo está organizado como segue: a seção 2 apresenta os trabalhos relacionados, a seção 3 apresenta o ambiente APSEE e o modelo proposto, a seção 4 apresenta uma visão geral da especificação e do protótipo desenvolvidos, e a seção 5 traz as considerações finais e trabalhos futuros.

2. Abordagens existentes e trabalhos relacionados

Durante o estudo realizado foram estudadas algumas das principais abordagens existentes na literatura especializada da área relativas a realização de mensuração de *software*, com destaque para três das principais abordagens existentes, o *Goal-Question-Metric* [14], *Goal-driven Software Measurement* [12] e o PSM - *Practical Software & System Measurement* [13]. Outros trabalhos relacionados abrangem experimentos e relatos de experiências realizadas com a utilização destas abordagens, além de outras variações. Alguns exemplos são as pesquisas de [3], [8], [15], entre outros. Outras pesquisas concentram-se em apresentar

novas abordagens para realização de mensuração e ferramentas de apoio. Alguns exemplos são [2], [5], [6], [7], [10], [16], [20] entre outros.

3. Modelo de mensuração

O modelo proposto neste trabalho tem como principal objetivo *prover suporte ao planejamento e execução de programas de mensuração em processos de software*, envolvendo atividades como: definição das métricas, modelagem do processo de mensuração, coleta, visualização e análise das métricas obtidas, avaliação do programa de mensuração, bem como a integração destas atividades com o processo de desenvolvimento. A abordagem proposta prevê a definição de um modelo de mensuração integrado a um ambiente de desenvolvimento de *software* orientado a processos (PSEE)², o APSEE, descrito na próxima seção. O modelo propõe a utilização de algumas das práticas propostas pelo PSM [13].

3.1 O ambiente para Automação de Processos de Software – APSEE

O ambiente APSEE é uma infra-estrutura de *software* que evoluiu a partir de um mecanismo de gerência de processos de *software* proposto em [17], com objetivo de definir um meta-modelo para a execução automatizada de processos de *software*. Atualmente, sua arquitetura evoluiu, sendo introduzidos novos recursos para apoiar de forma mais adequada as características evolucionárias e dinâmicas do processo de *software*, dando origem ao modelo chamado APSEE [18]. Os principais componentes do APSEE são organizados em três camadas: interação com usuário, mecanismos para gerência de processos, e repositórios. Uma breve descrição dos componentes do APSEE (fig.1) é apresentada a seguir, sendo que detalhes sobre as características e relacionamentos entre os componentes podem ser encontrados em [18].

O meta-modelo APSEE apresenta os tipos de dados utilizados pelo ambiente, abrangendo modelos de processo de *software* em seus diferentes estados, *templates* para reutilização, hierarquias de tipos relacionados aos componentes do APSEE, modelos de recursos de apoio utilizados pelas atividades, artefatos de *software*, regras definidas pelo usuário em relação aos modelos de processo, além de conhecimento sobre o processo, que permite definir e armazenar métricas e estimativas para os componentes do processo.

² PSEE - *Process-centered Software Engineering Environment*

A camada de mecanismos para gerência de processos descreve um conjunto de serviços que constituem o componente denominado *APSEE-Manager*, composto por elementos que abrangem uma linguagem para modelagem de processos (APSEE-PML), mecanismos de execução do processo, instanciação de processos através do uso de políticas de instanciação definidas pelo usuário, bem como mecanismos de apoio à reutilização de processos.

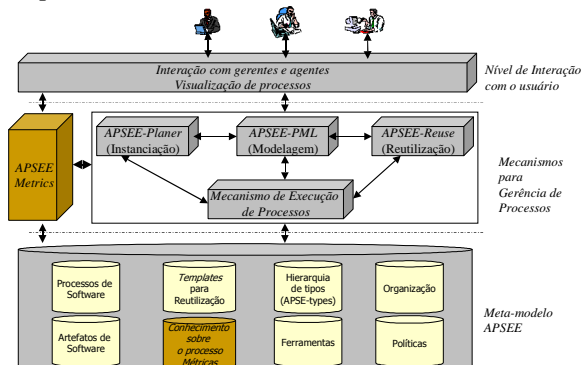


Figura 1. Evolução do modelo APSEE

A camada de interação é responsável pela coordenação da interação entre o usuário e o sistema. É através dela que o usuário se comunica com o sistema, avaliando o estado de um determinado processo, percebendo a evolução dinâmica das entidades que compõem o processo e intervindo quando necessário [19].

Na evolução proposta neste trabalho, o componente *APSEE-Metrics* foi integrado à camada de Gerência de Processos. Além disto, houve a necessidade de evolução do meta-modelo do APSEE, no que diz respeito ao gerenciamento das medições realizadas no processo de desenvolvimento, onde o componente *Conhecimento sobre o Processo* foi originalmente proposto para definir e armazenar métricas e estimativas para os componentes do processo, as quais podem ser consultadas dinamicamente durante a execução do mesmo. A evolução deste componente em relação às métricas está sendo proposta, de forma a permitir a definição e armazenamento de programas de mensuração de uma forma mais abrangente.

3.2 Mecanismo para gerenciamento das mensurações

O componente *APSEE-Metrics* consiste em um mecanismo responsável pelo gerenciamento da mensuração, de forma integrada ao ambiente e interagindo com os demais componentes do modelo. Buscando o atendimento das principais atividades relativas a mensuração em processos de *software*, o *APSEE-Metrics* foi organizado logicamente em quatro

sub-componentes principais: Planejamento, Coleta, Análise e Controle, e Avaliação. Estes componentes são apresentados na figura 2, e descritos em maiores detalhes na seqüência.

Planejamento: responsável pelo suporte ao planejamento da mensuração, abrangendo desde a definição do processo de mensuração até a definição do modelo de informação a ser utilizado, que contempla a definição e especificação das métricas, dos procedimentos de coleta e de análise dos resultados, resultando em um plano de mensuração.

Coleta: este componente é responsável pelo suporte a coleta dos dados das métricas definidas na etapa de planejamento, utilizando como base o plano de mensuração.

Análise e controle: este componente suporta a visualização das informações coletadas, bem como o registro das análises realizadas e ações efetuadas com base nestas análises.

Avaliação: compreende o armazenamento das experiências obtidas com a avaliação das métricas utilizadas, das atividades do processo de mensuração, além do próprio plano de mensuração, proporcionando um *feedback* sobre o programa de mensuração utilizado, servindo de apoio para novos planejamentos relativos às mensurações.

De forma simplificada, a sistemática de mensuração se inicia quando o especialista define um *plano de mensuração*, definindo informações relativas ao contexto do projeto, do processo de desenvolvimento utilizado e informações relativas à organização onde o processo está inserido. Com base nestas informações, o especialista pode escolher um *programa de mensuração* adequado às necessidades do projeto, ou definir um novo programa de mensuração. Fazem parte do programa o modelo de informação e o processo de mensuração.

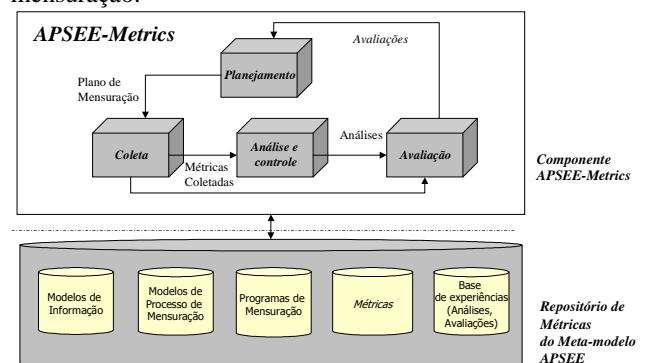


Figura 2. Componentes do APSEE-Metrics

No **modelo de informação** são definidos os **objetivos** da mensuração e as **informações críticas do projeto** e, baseado no modelo ICM [13] proposto pelo PSM, o gerente pode derivar e especificar as **métricas** necessárias para prover as informações relativas aos objetivos definidos. Também são definidas as **análises** a serem realizadas visando o acompanhamento do projeto, bem como as **avaliações** a serem realizadas ao final do processo. Para definição do modelo de informação as experiências anteriores armazenadas no repositório poderão ser utilizadas.

No **processo de mensuração** são definidos os procedimentos (atividades) de planejamento, coleta, análise e avaliação das métricas. Além disto, as atividades do processo de mensuração são integradas ao processo de desenvolvimento, sendo modeladas através dos recursos providos pelo ambiente APSEE para gerenciamento de atividades em um processo de *software*.

Após o planejamento, com base no plano de mensuração, a execução do plano é então seguida, executando-se o processo de mensuração. Conforme as atividades previstas no processo, os dados necessários para obtenção das métricas definidas são **coletados** e **armazenados**. Além disto, o gerente poderá ainda realizar o **acompanhamento** das métricas obtidas e, com base nestas informações, executar as **análises** previstas no processo de mensuração, armazenando as ações tomadas e os seus resultados, proporcionando base para tomada de decisão em futuros projetos.

Por fim, **avaliações** das métricas utilizadas, das atividades do processo de mensuração e do programa como um todo são armazenadas, visando a qualificação dos programas de mensuração e verificação de sua adequação ao contexto utilizado, permitindo a criação de uma base de informações úteis para criação de novos programas, ou de sua utilização em contextos similares.

3.3 Meta-modelo

Nesta seção são apresentadas as principais características do meta-modelo APSEE-*Metrics*, incorporado ao componente Conhecimento do Processo, originalmente proposto no APSEE.

O componente *ProcessKnowledge* (Conhecimento do Processo) funciona como base de conhecimento sobre os componentes do meta-modelo e foi integrado para ser consultado nas fases de modelagem, instanciação, simulação e execução, além de permitir estudos acerca do impacto de algumas métricas em aspectos importantes, como por exemplo, na qualidade do software produzido. Este trabalho propõe a inclusão de

novos recursos ao componente, que permitirão o gerenciamento das mensurações realizadas no ambiente APSEE.

A figura 3 apresenta o diagrama de classes em UML referente ao pacote de gerenciamento de mensurações (*MeasurementManagement*), introduzido no componente *ProcessKnowledge*, para permitir a definição e armazenamento de programas de mensuração. Através de seus componentes, o especialista poderá realizar a definição adequada de um plano de mensuração para cada projeto, realizando a definição das métricas apropriadas para cada contexto. A descrição e os relacionamentos entre as classes do modelo são descritos da seguinte forma:

Primeiramente, o especialista precisará definir um **programa de mensuração** (*MeasurementProgram*), definindo para o mesmo um identificador, um nome e a descrição do mesmo, além do contexto ao qual o programa poderá ser aplicado. Esta informação será bastante útil para permitir a reutilização do programa em outros projetos que sejam relacionados à contextos similares. Cada programa de mensuração é composto por um modelo de informação (*InformationModel*) e um processo de mensuração (*MeasurementProcess*).

O modelo de informação corresponde basicamente a definição das informações necessárias para a correta execução da mensuração, e é composto por:

Objetivos da mensuração (*MeasurementGoals*): correspondem a uma das principais informações do modelo, onde o gerente deverá definir as necessidades de informação do programa de mensuração. Os objetivos são compostos por um identificador, nome, descrição dos objetivos e resultados esperados com a mensuração. Além disto, cada objetivo pode ser priorizado, através de uma escala numérica que indica sua relevância em relação aos demais.

Informações críticas do projeto (*ProjectIssues*): uma vez obtidos os objetivos, são definidas informações críticas específicas do projeto (riscos, critérios, necessidades de informação do projeto, etc.), as quais são mapeadas para uma área comum de informação do modelo ICM³ (*CommonIssues*). Cada informação crítica do projeto possui um identificador, nome, descrição com detalhes sobre a informação, e também a probabilidade de ocorrência da mesma, bem como o impacto de sua

³ ICM (Issue-Category-Measures) – Modelo proposto pelo PSM (*Practical Software and System Measurement*) para guiar a seleção das medidas [13].

ocorrência em relação ao projeto, expressas em uma escala numérica.

Modelo ICM: no modelo ICM, são definidas áreas comuns que podem ser utilizadas como guia para definição das medidas apropriadas. Estas áreas são representadas neste modelo pela classe *CommonIssues*.

Cada uma destas, por sua vez, é composta por categorias de mensuração representada pela classe *MeasurementCategory*. Cada categoria possui medições genéricas (*GenericMeasures*), sendo que estas são mapeadas pelos objetivos de mensuração definidos anteriormente.

Cada uma das medidas genéricas é composta por métricas, que indicam quais os dados devem ser coletados para atender aos objetivos de mensuração definidos.

Métricas: são representada pela classe *Metric*, originalmente proposta no meta-modelo do APSEE. Cada métrica possui uma definição, representada pela classe *MetricDefinition*, e um tipo de métrica *MetricType*. As métricas obtidas são armazenadas pela classe *MetricCollected*, uma instância das métricas (*Metric*).

De acordo com a definição original do componente [18], a classe *MetricDefinition* permite definir métricas importantes para o processo e para a organização enquanto que as classes *MetricsCollected* e *Estimations* armazenam instâncias de métricas associadas aos componentes do modelo.

Análises: representadas pela classe *Analysis*, as análises a serem realizadas, com base nas métricas coletadas durante o processo de desenvolvimento, deverão ser especificadas durante o planejamento. Cada análise é relacionada com um ou mais objetivos de mensuração definidos. Também permite o armazenamento das experiências adquiridas durante a mensuração, sendo que a respectiva classe prevê uma identificação, um nome, uma descrição da análise a ser realizada (definida no planejamento), e os resultados das análises realizadas.

Avaliações (*Evaluation*): visando contemplar as avaliações necessárias ao programa de mensuração, é proposto neste modelo a definição e armazenamento de diferentes tipos de avaliações. A classe *Evaluation* é responsável por isto, e é definida por um identificador, um nome, uma descrição da avaliação a ser realizada (definida no planejamento das mensurações). Cada avaliação prevista poderá ter um tipo específico e o seu resultado. Os tipos podem denotar: a **avaliação das**

métricas utilizadas, a **avaliação das atividades** de mensuração ou a **avaliação do plano de mensuração**.

O processo de mensuração (*MeasurementProcess*) corresponde a definição das atividades necessárias para contemplar todas as etapas de mensuração. É composto por atividades de mensuração (*MeasurementActivity*), cada qual corresponde a um tipo especial de atividade, definida no meta-modelo APSEE, sendo que a mesma pode ser especializada em 4 tipos (definidos por *MeasurementActivityType*):

- a) Atividades de Planejamento das mensurações (*MAPlan*);
- b) Atividades de Análise (*MAAnalysis*);
- c) Atividades de Coleta das métricas (*MACollect*);
- d) Atividades de Avaliação (*MAEvaluate*).

Uma das características importantes deste modelo é a possibilidade de modelagem do processo de mensuração de forma integrada ao processo de desenvolvimento. A definição do processo de mensuração como uma especialização dos processos definidos pelo meta-modelo APSEE irá permitir que o mesmo possa ser integrado diretamente ao processo de desenvolvimento, trazendo vantagens como a modelagem, visualização, e outras funcionalidades já existentes no ambiente, permitindo ao gerente verificar o impacto da adoção de tais atividades no contexto geral de cada projeto.

Por fim, uma vez definido o programa de mensuração com seus objetivos, contexto, métricas, processo e demais informações, podemos relacionar um projeto (*Project*) do meta-modelo APSEE a este programa. Desta forma, é representado pela classe *MeasurementPlan*, o plano de mensuração do projeto.

O **plano de mensuração** é definido pela escolha de um dos programas de mensuração armazenados no modelo, além de informações gerais referentes a descrição textual do plano (contendo informações de relevância para o gerente do projeto, tais como fatores críticos de sucesso, considerações importantes sobre o projeto, etc.), o contexto ao qual o projeto está inserido (ambiente, tecnologias utilizadas, tipo de processo, etc.), além de outras informações adicionais relevantes. Estas informações servirão como base histórica para os projetos, relacionando experiências com a utilização de mensuração no ambiente APSEE.

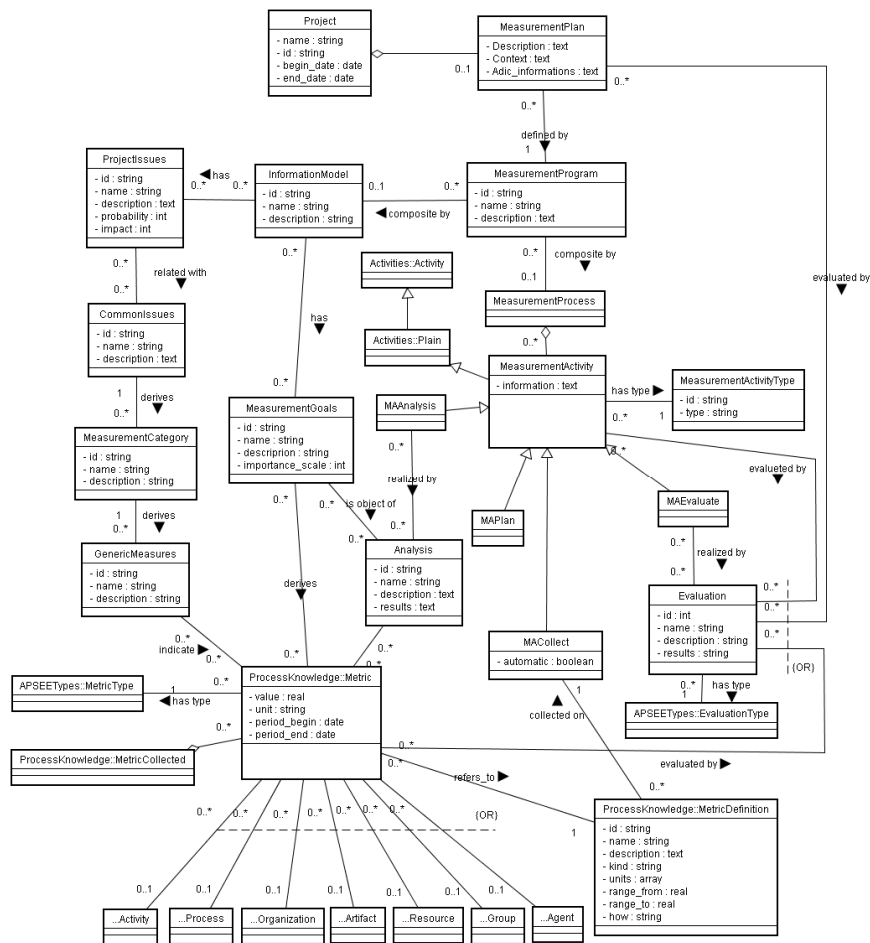


Figura 3. Modelo de mensuração

4. Especificação e protótipo do modelo

O modelo proposto foi especificado formalmente através de uma especificação algébrica. O formalismo adotado para a especificação formal do modelo proposto é denominado PROSOFT-Algébrico [18]. Este formalismo permite a descrição de tipos abstratos de dados através de um paradigma algébrico baseado em objetos. No PROSOFT-Algébrico, cada tipo de dado é instanciado a partir de um ATO (Ambiente de Tratamento de Objetos). Um ATO implementa um tipo abstrato de dados, e é composto por uma classe e um conjunto de operações que atuam sobre o objeto dessa classe. É importante ressaltar que o conceito de classe no PROSOFT é uma estrutura de dados que corresponde aos atributos do objeto, ou variáveis de instância nas linguagens orientadas a objeto.

A figura 4 ilustra a especificação algébrica da classe *InformationModel* que representa o modelo de informação das mensurações. As características de um

modelo de informação são definidas através de um mapeamento, onde o domínio da função é um identificador do modelo de informação, e sua imagem como sendo um registro contendo os atributos *Name*, *Description*, *Goals* (conjunto de objetivos de mensuração), *MetricDef* (conjunto de especificações das métricas a serem coletadas), *MetricsCollected* (conjunto de métricas coletadas) *Analyses* (conjunto de análises a serem realizadas), *Evaluations* (conjunto de avaliações a serem realizadas).

As demais classes do modelo e os detalhes relativos à especificação das operações e assinatura das mesmas são apresentadas em [21], e foram omitidas neste artigo em função da limitação de espaço. Baseado na especificação realizada, foi desenvolvido um protótipo de uma ferramenta de apoio à abordagem proposta, provendo apoio computacional a mesma, além de servir como uma maneira de validar o modelo.

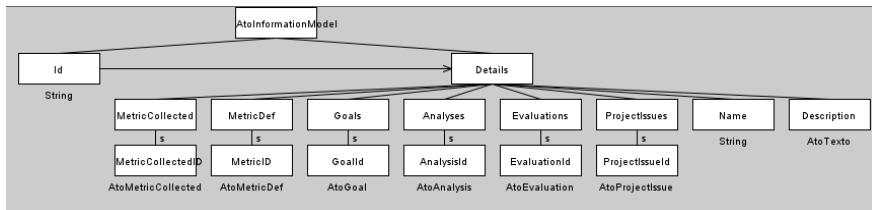


Figura 4. Especificação algébrica do modelo de informação

Um aspecto importante é que a implementação do modelo foi realizada através de um ambiente de execução de processos de *software*, um PSEE, uma vez que a ferramenta foi integrada ao ambiente de modelagem e execução de processos do APSEE-Prosoft. O protótipo contempla o registro de cada um dos elementos do modelo proposto, bem como a modelagem das atividades do processo de mensuração de forma integrada ao processo de desenvolvimento.

A figura 5 ilustra a modelagem do processo de mensuração, de forma integrada ao processo de desenvolvimento. As elipses representam atividades, interligadas através de conexões que representam suas dependências e os cubos representam os artefatos produzidos/consumidos por cada atividade. Estes elementos apresentados são modelados através da APSEE-PML [17].

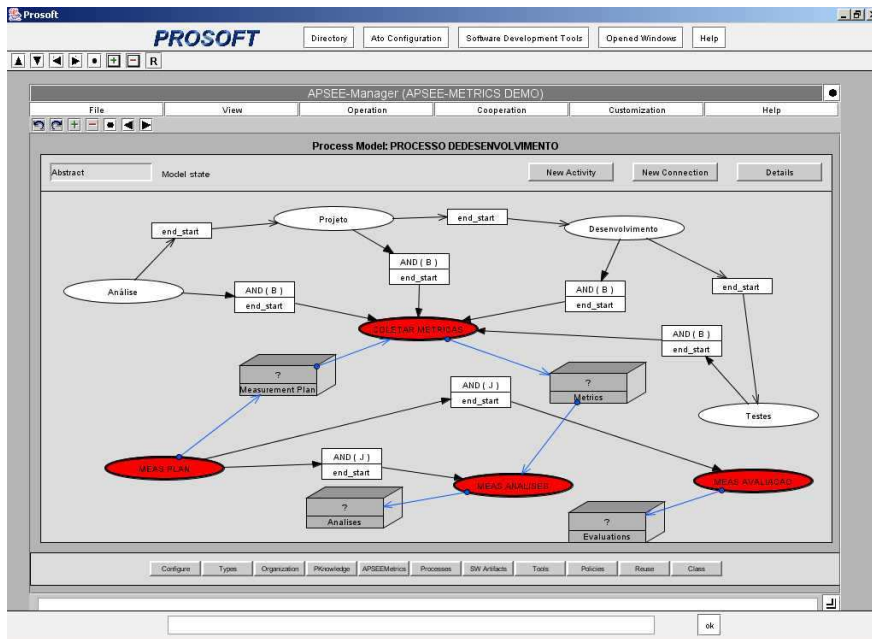


Figura 5. Modelagem do processo no APSEE-Metrics

5. Conclusões e trabalhos futuros

Este trabalho apresentou um modelo para mensuração em processos de *software*, o qual possibilita sua condução, de forma organizada, através da definição de programas de mensuração. O modelo proposto foi integrado ao ambiente APSEE de modelagem e execução de processos, fornecendo mecanismos para a definição dos programas de mensuração para os processos de *software* definidos no ambiente APSEE, com o objetivo de facilitar a condução das atividades

relativas à mensuração em projetos de desenvolvimento de *software*.

O modelo APSEE-Metrics foi especificado formalmente, com a utilização do formalismo Prosoft-Algébrico. A especificação formal deu origem à implementação de um protótipo, o qual foi aplicado a um cenário real de utilização em uma organização desenvolvedora de *software*. Embora o trabalho aqui apresentado não forneça uma solução completa para o problema investigado, acredita-se que o modelo

proposto constitui uma contribuição importante para a área de pesquisa, em virtude da especificação apresentada e da prototipação realizada. Além disto, a integração promovida com o ambiente APSEE proporcionou também um avanço importante do projeto Prosoft em relação ao ambiente de desenvolvimento de *software*, o qual, diversas pesquisas relacionadas às áreas de Engenharia de *Software* e Processos de *Software* vêm sendo desenvolvidas pelo grupo.

Além disto, o modelo proposto trouxe algumas contribuições para a área em estudo, podendo ser destacado: i) um conjunto relevante de requisitos elaborados com base em um estudo detalhado sobre as principais abordagens existentes e consolidadas na literatura especializada da área; ii) o modelo de mensuração viabiliza o planejamento dos elementos necessários para que se possa realizar mensuração em processos de *software*, contemplando as principais etapas necessárias para a condução de programas de mensuração; iii) o mecanismo de modelagem do processo de mensuração permite a documentação e facilita a análise e a gerência das atividades necessárias para cada uma das etapas deste processo. Além disto, a integração destas atividades com o processo de desenvolvimento é um fator de importante relevância, uma vez que permite a interação de diferentes aspectos relacionados à gestão de um processo de desenvolvimento de *software*, o que inclui necessariamente a mensuração; iv) os diferentes componentes envolvidos na definição do modelo foram especificados formalmente, constituindo uma base semântica de alto nível de abstração, o que traz muitos benefícios quando se considera a obtenção de uma especificação independente de implementação, tendo também o potencial de permitir que as idéias do trabalho possam ser implementadas em outros ambientes de desenvolvimento de *software*; v) um protótipo que implementa o modelo de mensuração foi construído no ambiente Prosoft-Java e integrado ao ambiente APSEE. Essa integração complementa ainda mais o APSEE, e permite que o APSEE-*Metrics* utilize os recursos disponíveis no ambiente.

À partir deste trabalho, visualiza-se o desenvolvimento de outros trabalhos que abordem aspectos complementares ao tema abrangido, tais como: i) a avaliação empírica da influência do modelo proposto na gerência de projetos de *software*. ii) a incorporação ao modelo da utilização de estimativas de projeto; iii) em relação ao protótipo, aspectos de usabilidade e facilidades em relação à relatórios e demais

funcionalidades necessitam ser aperfeiçoadas, como a coleta das métricas, bem como a apresentação das mesmas de forma mais interativa; iv) outro tópico para trabalhos futuros é a viabilização da coleta automática de métricas através do relacionamento de atributos de cada um dos componentes do ambiente APSEE (tais como atividades, pessoas, artefatos, etc.), de forma integrada com as métricas definidas no modelo.

Por fim, este trabalho possibilitou a combinação de aspectos relacionados às importantes áreas da Engenharia de *Software*, como mensuração, processos de *software*, ambientes de engenharia de *software* centrados no processos (PSEE), bem como a utilização de métodos formais para definição e especificação de um modelo de mensuração integrado a um PSEE.

Referências

1. MENS, T.; DEMEYER, S. Future trends in software evolution metrics. In: INTERNATIONAL WORKSHOP ON PRINCIPLES OF SOFTWARE EVOLUTION IWPSE, 4., 2001, Vienna, Austria, Proceedings... New York, NY: ACM Press, 2001. p.83-86.
2. FRANCA, L.P.A.; VON STAA, A.; LUCENA, C.J.P. de. Medição de Software para Pequenas Empresas: uma Solução Baseada na Web. Rio de Janeiro: PUC-RJ, 1998. 24 p.
3. PERKINS, T.; PETERSON R., SMITH, L. Back to the Basics: Measurement and Metrics. USA: Department of Defense and USA Army, 2003. Disponível em: <www.stsc.af.mil/crosstalk> Acesso em: out. 2005.
4. VAVASSORI, F. B. Metodologia para o Gerenciamento Distribuído de Projetos e Métrica de Software. 2002. 211 f. Tese (Doutorado) - UFSC, Florianópolis.
5. BASILI, V. R. et al. Lessons learned from 25 years of process improvement: the rise and fall of the NASA software engineering laboratory. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, ICSE, 24., 2002, Orlando, Florida. Proceedings... New York: ACM Press, 2002. p. 69-79.
6. BORGES, E.P. Um modelo de medição para processos de desenvolvimento de software. 2003. 154 f. Dissertação (Mestrado) - UFMG, Belo Horizonte.
7. GOETHERT, W.; FISHER, M. Deriving Enterprise-Based Measures Using the Balanced Scorecard and

- Goal-Driven Measurement Techniques. Pittsburgh: Carnegie Mellon University, 2003.
8. FUGGETTA, A. et al. Applying GQM in an industrial software factory. *ACM Transactions on Software Engineering and Methodology*, New York, v.7, n.4, p.411-448, Oct. 1998.
 9. TANAKA, T et. al Software quality analysis and measurement service activity in the company. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 20., Kyoto, Japan. Proceedings... Washington, DC: IEEE Computer Society, 1998. p. 426-429.
 10. OLSINA, L.A. et al. Un Marco Conceptual para la Definición y Explotación de Métricas de Calidad. In: JORNADAS DE INGENIERÍA DEL SOFTWARE Y BASES DE DATOS, 7., 2001, Málaga, Espanha. Anais... Málaga: University of Málaga, 2001.
 11. NUNES, D.J. PROSOFT: Um ambiente de Desenvolvimento de Software baseado no metodo algébrico. Porto Alegre: UFRGS, 1995. 20 p.
 12. PARK, R.E.; GOETHERT, W.B.; FLORAC, W.A. Goal-Driven Software Measurement: a Guidebook. Carnegie Mellon University, 1996. Disponível em: <http://www.sei.cmu.edu/publications/documents/96.reports/96.hb.002.html>.
 13. USA. Department of Defense and USA Army. Practical Software and Systems Measurement: A Foundation for Objective Project Management. Washington, D.C.: Department of Defense and US Army, 2003. Disponível em: <http://www.psmc.com>. Acesso em: 05 ago. 2005.
 16. KRISHNAN, M.S.; KELLNER, M.I. Measuring Process Consistency: Implications for Reducing Software Defects. *IEEE Transactions on Software Engineering*, Los Alamitos, CA, v.25, n.6, p.800-815, Nov. 1999.
 17. REIS, C.A.de L. Um gerenciador de processos de software para o ambiente Prosoft. 1998. 197 f. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
 18. REIS, C.A.de L. Uma Abordagem Flexível para Execução de Processos de Software Evolutivos. 2003. 215 f. Tese (Doutorado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
 19. SOUSA, A.L.R. de. APSEE-Monitor: um mecanismo de apoio a Visualização de modelos de processos de software. 2004. 112 f. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
 20. ANACLETO, A.; VON WANGENHEIN, C.G. Aplicando Mensuração em Microempresas de Software para suporte da Gerência de Projetos. In: SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE, 1., 2002. Anais... Gramado: SBC, 2002.
 21. GARCIA JÚNIOR, P.R. APSEE-Metrics – um modelo para mensuração em processos de software. 2006. 165f. Dissertação (Mestrado em Ciência da Computação) Instituto de Informática, UFRGS, Porto Alegre.