



## **Especificação de métricas de arquitetura de linha de produto de software em Structured Metrics Metamodel**

Lailson Miranda Tofanelli (PIBIC/CNPq/FA/Uem), Edson A. Oliveira Junior (Orientador), e-mail: lailsontofaneli@gmail.com; edson@din.uem.br

Universidade Estadual de Maringá/Departamento de Informática/Maringá, PR.

**Área de Conhecimento: 1.00.00.00-3 Ciências Exatas e da Terra, 1.03.00.00-7 subárea: Ciência da Computação, 1.03.03.02-2 subárea: Engenharia de Software.**

**Palavras-chave:** Linha de Produtos de Software, Métricas, SMM.

### **Resumo:**

Este trabalho é parte da definição de métricas de complexidade e extensibilidade já especificadas em *Open Core*. Entretanto, a OMG propôs um padrão para definição de métricas denominado SMM. Assim, é visto a necessidade de padronizar as métricas de complexidade e extensibilidade para PLA com base no SMM.

### **Introdução**

Para avaliar uma Linha de Produto de Software (LPS) pode-se analisar os potenciais produtos a serem produzidos a partir da arquitetura de LPS (*Product Line Architecture - PLA*). Assim, a avaliação de PLA pode ser vista como um meio de avaliar uma LPS considerando seus atributos de qualidade. Para tanto foram estabelecidas métricas de complexidade e extensibilidade (OLIVEIRAJR et al., 2010). Tais métricas foram especificadas com base no *framework Open Core*. Porém, a *Object Management Group* (OMG) criou um padrão para definição de métricas com base em um metamodelo, chamado *Structured Metrics Metamodel* (SMM) (OMG, 2015). Portanto, este trabalho tem como objetivo especificar as métricas de complexidade e extensibilidade para PLA com base no metamodelo SMM.





## Materiais e métodos

Inicialmente, foi realizada uma revisão bibliográfica sobre LPS, PLA e variabilidades. Em seguida, foram estudadas as métricas de complexidade e extensibilidade existentes. Foi realizado também um estudo completo do metamodelo SMM, como especificar uma métrica em SMM e os passos para a execução de um processo de medição. Após isso as métricas de complexidade e extensibilidade foram especificadas em SMM e, então, aplicadas em um exemplo de medição e coleta de valores seguindo os conceitos do SMM. Após isso, foi realizado uma comparação com as métricas em SMM e com as métricas em *Open Core*.

## Resultados e Discussão

O *Structured Metrics Metamodel* (SMM) visa padronizar a definição das métricas e representar os resultados das medições, facilitando a troca de dados (OMG, 2015). A especificação de métricas em SMM deve levar em consideração o que ela vai medir e como. Assim, define-se o tipo da métrica e a biblioteca (*MeasureLibrary*) que ela irá pertencer. Nesse trabalho foram definidas duas *MeasureLibraries*, uma para as métricas de complexidade e uma para as de extensibilidade. A escolha do tipo de cada métrica implica em atributos diferentes, e consecutivamente, o processo de medição será diferente em alguns pontos.

Cada métrica terá seus atributos básicos, como *Name*, *Description*, *Characteristic*, *Unit*. E também o atributo principal de uma métrica SMM que irá ditar o rumo da medição, sendo ele o *Operation*, que define o código que será aplicado ao modelo alvo. Para as 12 métricas que compõem os conjuntos das métricas de complexidade e extensibilidade foram criados um *Operation* para cada métrica. Visto isso, o código *Operation* para a Métrica *ExtensClass* é apresentado na Figura 1.

: Operation
- Name : ExtensClassOperation - Language : XPath - Body : //packagedElement and [@xmi:type='uml:Class'] and number//count((/child:ownedOperation and [@xmi : type = 'uml:Operation'] and [@sAbstract='true'])) div //count(/child:ownedOperation and [@xmi : type = 'uml:Operation'])

Figura 1 – Código *Operation* da Métrica *ExtensClass*.

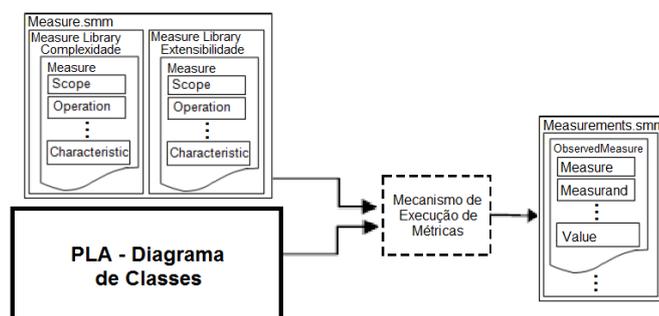
O *Operation* da *ExtensClass* é interpretado da seguinte forma: procure em todo o arquivo XMI, um elemento cuja *tag* tem o nome de “*packagedElement*” onde seu tipo de elemento “*xmi:type*” seja “*uml:Class*” e





retorne um valor “*number*” sendo a divisão entre a quantidade de métodos abstratos identificado por um atributo “*isAbstract*” com valor *True*, e pela quantidade de métodos da classe, retornada pelo “*count*”, ou seja, a quantidade de nós retornados. Cada método é definido por um atributo “*xmi:type*” com valor “*uml:operation*”. Cada consulta irá retornar para uma determinada classe um valor referente a sua extensibilidade. O código *Operation* não pode ser considerado totalmente correto, pois sem uma ferramenta para análise e consulta em um modelo alvo, o código se torna apenas uma abstração do que deve ser realizado na medição.

Para o processo de medição SMM, o modelo padrão utilizado neste trabalho (*ObervationScope*) é um diagrama de classes, criado na ferramenta *AstahProfessional*, e todas as métricas foram especificadas com base nesse modelo XMI. Na Figura 2 é explicado o projeto como um todo.



**Figura 2** - A: Especificação das métricas em SMM (Adaptado de HONDA, 2014); B: Modelo alvo para aplicação das métricas; C: Resultado da medição.

O estudo de caso foi realizado com base em um estudo analítico dos dois tipos de especificação de métricas. O *Open Core* devido a uma ferramenta já disponível para seu uso, conta com mais segurança e facilidade de uso que o SMM, que durante a realização do trabalho as especificações de métricas e a execução da medição foi feito manualmente, o que possibilita maiores chances de erros. O SMM também é muito complexo para ser adotado em um projeto, pois é necessário que se estude todo o seu metamodelo de forma ampla.

## Conclusões

As definições de cada métrica em SMM permitem que suas características sejam totalmente compreendidas, entretanto, realizar a especificação de





uma métrica considerando o SMM, se trata de uma atividade complexa. E a maior dificuldade é em relação ao código *Operation*. Este código não é totalmente eficiente, ele apenas permitiu uma abstração do que é necessário ser realizado, pois sem uma ferramenta para a medição, não é possível constatar se o *Operation* realmente cumpre com o que está definido. Assim, um trabalho futuro proposto é a criação de um ambiente, composto por um analisador que especifique e processe métricas em SMM

### Agradecimentos

Agradecemos ao Programa Institucional de Bolsas de Iniciação Científica PIBIC/CNPq-FA-UEM por incentivar e financiar a realização deste trabalho.

### Referências

BERA, M. H. G.; OLIVEIRAJR, E.; COLANZI, T. E. **Evidence-based SMarty Support for Variability Identification and Representation in Component Models**. In: 17th International Conference on Enterprise Information Systems (ICEIS), 2015, Barcelona, Espanha. 2015. v. 2. p. 295-302.

HONDA, Raphael. **Modelagem e Cômputo de Métricas de Interesses no Contexto de Modernização de Sistemas Legados**. 2014. 119 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Ciência da Computação, Universidade Federal de São Carlos, São Carlos, 2014.

LINDEN, F. J.; SCHMID, K.; ROMMES, E. (2007) **Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering**. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

OLIVEIRAJR, E.; GIMENES, I. M. S.; MALDONADO, J. C.; MASIERO, P. C.; BARROCA, LEONOR. **Systematic Evaluation of Software Product Line Architectures**. Journal of Universal Computer Science (JUCS), v. 19, p. 25-52, 2013.

OMG (2015). **Structured Metrics Metamodel Specification v1.1**. Disponível em <http://www.omg.org/spec/SMM/1.1>. Acessado em 15/06/2016.

