

# 软件度量的概念模型<sup>\*</sup>

侯红 刘晓霞 郝克刚

(西北大学计算机科学系 西安 710069)

**摘要** 度量面临的重要问题是建立引入和运用度量的一个有效的方法和框架。使用统一的术语、明确的语义对组织的过程模型进行定义;创建具体的度量模型参考框架的度量元模型,建立组织整体的度量管理。为解决上述问题,本文建立了软件度量的概念模型 CMMP (Conceptual Model of Measurement Process),它是基于软件过程工程元模型 (SPEM)的,从而可与现有的过程模型有效地集成。

**关键词** 软件度量,软件过程改进,软件过程工程元模型

## Conceptual Model of Measurement Process

HOU Hong LIU Xiao-Xia HAO Ke-Gang

(Computer Science Department of Northwest University, Xi'an 710069)

**Abstract** The important issue of software measurement is to establish and apply an effective method and framework; Using the unified terminology and definite semantics to define the organizational process model, and establish concise measurement meta-model and build the organizational integrated measurement management. To solve the above problem, CMMP (Conceptual Model of Measurement Process) based on the software process engineering meta-model is proposed in this paper. It can be integrated with the available process model.

**Keywords** Software measurement, Software process improvement, SPEM

### 1 软件度量与软件过程建模

软件度量只有在它是一个稳定的软件管理规则的一部分时才可能是高效的<sup>[1]</sup>。只有度量稳定且已定义的过程,软件组织才能从度量中获得较多的有意义的信息。定义组织的软件过程是进行过程度量的前提。同样,由于过程度量本身也是一个过程,它也必须是一个已定义的、稳定的过程。

目前,在软件工程实践中进行过程度量都会处于下列3种情况之一:

- (1)过程没有定义,过程度量以文字的形式定义。
- (2)过程用自然语言定义,但过程度量仍以文字形式定义。
- (3)过程已有半形式化或形式化的定义,过程度量的定义与过程定义相结合,作为过程定义的扩展。

其中,在第一种情况下,由于文字上的实体定义和过程环境下的实体定义可能不同,会造成度量的定义和它在实际过程环境下的实施不一致。第二种情况下,这种不一致仍可能存在,但其可能性已大大减少,因为对过程的定义已经在一定程度上对一致性进行了检查。而第三种情况就能较好地消除过程定义和过程度量定义之间的歧义,使度量计划的实施与其定义一致。

在将软件过程模型和过程度量模型结合方面已经有不少研究。例如,Matsumoto 等人利用 Petri 网技术定义过程模型及数据模型。Alfred Brockers 和 Christiane Differding 研究了如何在保持软件组织已有的开发模式基础上利用过程建模的技术引入软件度量计划<sup>[2]</sup>。Maurizio Morisio 利用面向对象的概念对过程进行建模,并在过程模型基础上定义度量的目标和过程,并实现该度量<sup>[3,4]</sup>。此外, Richard Webby 和 Ulrike Becker 引入了一种逻辑模式来促进对各种概念和术

语的公共理解,以联接过程建模和软件度量两个不同领域<sup>[5]</sup>。文[6~8]将元模型的思想用于软件过程的研究中。本文在借鉴文[8]思想的基础上,将元模型用于软件度量过程的研究。

元模型是描述模型构成的语言,而过程元模型是用于表示和组织软件过程模型的概念框架。MDA 框架的核心是4层元建模架构,如表1所示。在4层架构中,上一层是用来描述下一层的语言,位于最高层 M3 层的是元对象设施 MOF (meta object facility)<sup>[9]</sup>,又称为元元模型。MOF 是自描述的。基于 MOF 可以面向不同的领域定义不同的描述语言,即 M2 层的元模型。

由于不同的过程具有不同的特性,要解决过程模型一致的根本问题,必须建立统一的过程元模型。为此,OMG 提出了软件过程工程元模型 SPEM。SPEM 提取了 RUP 等多个软件开发过程中的公共特征,是一种软件过程的通用元模型。由于植根于 UML 概念,SPEM 易于为广大开发者和建模者所理解。同时,SPEM 支持 MDA 框架,提供模型和工具的互操作性。因此,选择 SPEM 作为软件度量过程元模型的研究基础。

### 2 软件度量的信息模型

度量信息模型是将已定义的信息需要与实际要度量的项目软件过程、产品和实体联系起来的机制。它为相关的度量概念建立了一个已定义的结构,而且在组织内准确地交流度量结果提供了基础。图1是有关度量信息模型的关系<sup>[10]</sup>。

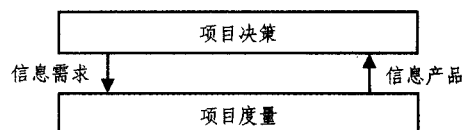


图1 度量信息模型的关系

<sup>\*</sup> 本文受国家 863 计划(编号:2004AA115090)资助。侯红 副教授,博士,主要研究方向:软件过程工程、软件度量;刘晓霞 副教授,主要研究方向:智能信息处理;郝克刚 教授,博士生导师,主要研究方向软件工程与理论。

一般情况软件度量的信息需求可分为七类<sup>[9]</sup>。

**进度和进展:**这个信息分类是针对项目里程碑的实现以及个人工作单元的完成。落后于进度的项目要想满足它的交付目标,通常只能削减功能或是牺牲产品质量。

**资源和成本:**这个信息分类是关于在被执行的工作与分配给项目的人力资源之间的平衡。超出预算工作量的项目通常只能通过削减软件功能或牺牲产品质量来进行补偿和恢复。

**产品规模和稳定性:**这个信息分类针对功能的稳定性以及要求的软件能力。该信息分类还与交付的用来提供所需能力的软件容量有关。稳定性包括功能范围或数量的变更。软件规模的增加通常要求增加所用的资源或是延长项目的进度计划。

**产品质量:**这个信息分类针对已交付的软件产品无故障地支持用户需要的能力。若交付了低质量的产品,则使其正常运转的责任通常落在负责维护的组织身上。

**过程性能:**这个信息分类涉及与项目需要相关的供应商的能力。缺乏软件开发过程或低生产率的供应商,可能难以满足积极进取的项目进度和成本目标。

**技术有效性:**这个信息分类针对所建议的技术方法的可行性。它针对工程方法,如软件复用、商业软件构件的使用、对高级软件开发过程的依赖以及通用软件构架的实现。如果已提出的技术方法中的关键元素无法实现,则可能导致成本的增加和进度的拖延。

**客户满意度:**这个信息分类针对项目交付的产品与服务满足客户期望的程度。满意度的指示器可以从客户的反馈和所要求的客户支持的级别获得。

### 3 软件度量的概念模型 CMMP

软件过程管理由于涉及到许多方面,因此有必要建立一个统一的概念模型。概念模型框架有机地将软件过程管理和软件度量结合在一起,以进行软件过程改进。

对于一个组织来说,为了实现软件过程的统一管理,有必要为下面两点建立一个基础<sup>[8]</sup>:使用统一的术语、明确的语义对组织的过程模型进行定义;采用作为创建具体的度量模型参考框架的度量元模型,建立组织整体的度量管理。

为了统一表述软件过程的上述两个重要方面,研究目标是在 MDA 框架下建立软件度量过程的概念模型——CMMP (Conceptual model of measurement process),这就要求 CMMP 必须满足以下两个条件:CMMP 应当符合 MOF, MDA 框架为基于 MOF 的各种模型提供模型转换和数据交互的支持,其首要条件是所有的建模语言都应当符合 MOF; CMMP 应当支持 ISO15939 软件过程建模,既要体现软件度量特定过程语义,也要具备表示可操作过程的能力。CMMP 抽象为四个层次,见表 1。

表 1 MOF(Meta Object Facility)概念模型级别及其应用环境

级别	MOF	应用环境
M3	MOF 模型 元-元模型	MOF 模型
M2	元模型	软件过程工程元模型(SPEM) 通用测量元模型(ISO 15939)
M1	模型	具体的过程模型 具体的度量模型
M0	数据	过程模型的实例 度量模型的实例

最底层 M0 包括的结果有:

(1)过程模型的应用,如某一具体项目的评估与改进模型、维护模型。在该级别具体项目的执行结果被记录。

(2)度量过程的应用,遵循某一度量模型所收集到的具体的数据被记录,如所度量到的关系数据库的值或 UML 类图的值等。

在 M0 所管理的数据是用 M1 所表示的数据的实例。在 M1 中,按照文中给出的概念框架,包含了软件过程定义的具体模型和其度量模型。从定义的观点看,在这一级别给出公司的过程模型,如开发、维护、评价和改进过程的模型等。从度量的观点看,这一级别包括公司使用的度量模型,如关系、数据库的度量等和具体的度量软件工件如 UML 类图、状态变迁图等。同时,在 M1 公司还应该建立定义的过程模型本身的度量模型。

所有在 M1 定义的模型是 M2 所表示概念的具体实例。因此,在 M2 是对概念模型的抽象,应包括创建具体模型的通用元模型。在本框架中的通用元模型有:

(1)软件过程元模型,可用它来定义具体的过程模型。

由于被工业界广泛采纳,将 SPEM 作为软件过程的元模型。该元模型包含了用来定义具体的软件过程模型的创建器。

(2)度量元模型,可用它来定义具体的度量模型。

在该框架概念模型的最后一级(M3),所有的过程元模型和度量元模型的概念被表示出来。使用 MOF 抽象语言来定义,MOF 抽象语言由两部分构成:MOF 类和 MOF 关联。按照这种方式,在 M2 级中的所有概念是 MOF 类或 MOF 关联的实例。

在度量过程中,收集数据的主要问题可归结为软件测量元低劣的定义。因此,不仅需要获取有效地忠实于度量过程的数据,而且需要有效地表示与这些数据相关的元数据。因此,非常有必要建立一种通用的度量元模型,可派生出建立组织过程评估和改进基础的具体度量模型<sup>[8]</sup>。图 2 是基于 ISO15939 标准<sup>[11]</sup>给出的用 UML 表示的度量元模型。

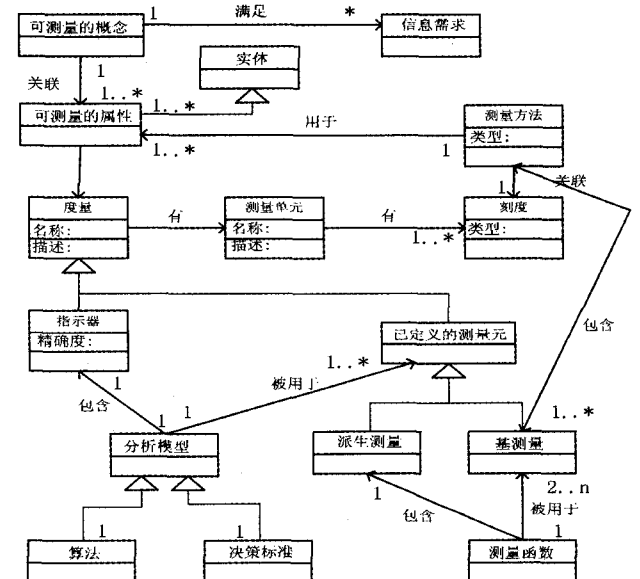


图 2 度量过程的通用元模型

从图 2 可看出,可被测量的属性元素为实体。可测量的属性是一个软件实体的可区别的特性或特征。实体包括过程、产品、项目和资源。属性的目标是满足特定的信息需求,如“比较软件生产率是否达到了一个确定值的需要”,在属性

和信息需要间的关系的抽象通过可测量的概念这一元素表达。在上面的例子中,应当是“软件开发的生产率”。作为可测量的属性,将采用开发产品的规模和开发的工作量表示。

所有可测量的属性与度量相关,度量是用于量化的不同类型测量的进一步抽象,并为相关实体进行决策。所有度量均有隶属于某一度量刻度下的测量单元<sup>[12]</sup>。三种度量类型为:

(1)基测量

基测量的数学描述为:  $y=f(x)$ , 其中  $y$  为基测量目标值,  $f$  为测量方法,  $x$  为属性。测量方法由一系列的逻辑运算构成,通常用于量化某一刻度下的属性<sup>[13]</sup>,如统计某时间段内的数量或经历的时间等。

(2)派生测量

软件派生测量的数学模型为  $z=g(y_1, y_2, \dots, z_1, z_2, \dots)$ , 其中  $z$  为目标派生测量值,  $g$  为测量函数,  $y_1, y_2, \dots$  为基测量值,  $z_1, z_2, \dots$  为派生测量值。测量函数是在两个或多个基测量基础上的算法或运算<sup>[13]</sup>。

(3)指示器

指示器的形式化描述:

$$\begin{cases} W=h(y_1, y_2, \dots, z_1, z_2, \dots) \\ w \in g, \text{其中 } g \text{ 为决策准则或决策条件} \end{cases}$$

其中  $W$  为指示器目标值,  $h$  为度量分析模型,  $y_1, y_2, \dots$  为基本度量值,  $z_1, z_2, \dots$  为派生测量值<sup>[13]</sup>。

分析模型产生与已定义的信息需求相关的估算或评价。它由与确定的相关的决策标准结合了一个或多个基测量与/或派生测量的算法或计算得到。所有的决策标准由一系列的限值、确定研究需要所使用的对象或达到确定结果相关的可信度构成。这些标准用于解释度量结果。

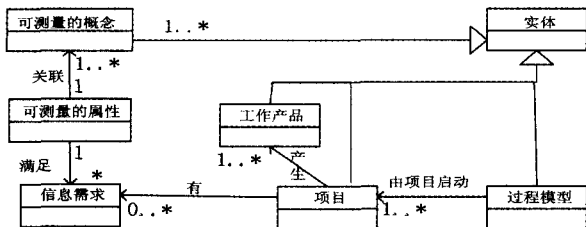


图3 软件过程元模型和度量元模型间的关系

使用图2给出的元模型可度量过程或数据模型的任何元素。图3给出软件过程元模型和软件度量元模型的主要关

系。从图3我们可以看出,任何过程模型都是在具体的项目下执行的。作为软件项目执行的结果,产生工作产品且所有的软件项目需要满足某些信息需求。

**小结** 创建度量的过程模型是软件度量过程的难点之一,本文对度量的概念模型研究的过程中,为保证术语的一致性和度量的可复用性,采用已有的度量标准 ISO15939 作为基础。通过实践证明度量的过程概念模型的建立有助于减少对度量的误解,提高度量的有效性。

参考文献

- 1 Koch F J. Metrics and the Immature Software Process. 2000. <http://www.qpmg.com/metrics.htm>
- 2 Brocker A, Differding C. The Role of Software Process Modeling in Planning Industrial Measurement Programs. In: Proceedings of the 3rd International Metrics Symposium, Berlin, March 1996
- 3 Morisio M. A methodology to measure the software process. In: Proceedings of the 7th Annual Oregon Workshop on Software Metrics, 1995
- 4 Aarsten A, Morisio M. Using object oriented technology to measure a software process. In: Proceedings of Achieving Quality in Software (AQUIS96), Florence, Italy
- 5 Webby R G, Becker U. Towards a Logical Schema Integrating Software Process Modeling and Software Measurement. In: Proceedings of the Process Modeling and Studies of Software Evolution Workshop, ICSE97, ACM and IEEE, Boston, May 1997
- 6 李娟, 李明树, 武占春, 等. 基于 SPEM 的 CMM 软件过程元模型. 软件学报, 2005, 16(8)
- 7 李娟, 袁峰, 李明树. 一种基于模型融合的 CMM 实施过程建模方法. 计算机学报, 2006(1)
- 8 García F, Ruiz F, Cruz J A, et al. Integrated Measurement for the Evaluation and Improvement of Software Processes, 2003
- 9 Object Management Group. Meta object facility (MOF) specification. Version 1. 4, formal/02-04-03; Needham, Object Management Group, 2002. <http://doc.omg.org/formal/02-04-03>
- 10 McGarry J, Card D, Jones C, et al. Practical software Measurement; Objective Information for Decision Makers, Pearson Education, Inc, 2001, 10
- 11 International Organization for Standard. Software Engineering : Software measurement process, ISO/IEC 15939. 2002(E)
- 12 侯红, 郝克刚. 软件测量刻度及选择方法. 计算机科学, 2005, 32(5)
- 13 李兴南. 软件测试度量的研究及其工具 STMT 开发. [硕士论文]. 西北大学, 2005

(上接第 215 页)

的改进算法可得约简为  $\{b, c\}$ , 根据我们提出的算法 2, 也可得到该不一致决策表的约简为  $\{b, c\}$ 。

**结论** 本文所提出的划分贴近度理论是基于粗糙集的集合思想, 所提出的属性约简算法 1 能够对一般信息系统进行有效的约简, 算法 2 对一致和不一致决策信息系统都能进行有效的约简, 并通过实例验证了两个算法的有效性。

参考文献

- 1 Pawlak Z. Rough Set-Theoretical Aspect of Reasoning About Data [M]. Kluwer Academic pub, 1991
- 2 Lingras P J, Yao Y Y. Data mining using extensions of the rough set model [J]. Journal of the American Society for Information Science, 1998, 49(5): 415~422
- 3 Li Y C, Fang T J. Rough Set Methods for Constructing Support Vector Machines [A]. In: 9<sup>th</sup> International Conference Rough

- Sets, Fuzzy Sets, Data Mining And Granular Computing [C]. Chongqing China, 2003, 334~338
- 4 Skowron A, Rauszer C. The discernibility matrices and functions in information [A]. Intelligent Decision Support Handbook of Applications and Advances of the Rough Sets Theory [C]. Kluwer Academic Pub, 1992. 331~338
- 5 Wang J. Reduction Algorithms Based on Discernibility Matrix The Ordered Attributes Method [J]. Jounary Computer Science Technology, 2001, 16(6): 498~504
- 6 苗夺谦, 胡桂荣. 知识约简的一种启发式算法 [J]. 计算机研究与发展, 1999, 36(6): 681~684
- 7 王国胤, 于洪, 杨大春. 基于条件信息熵的决策表约简 [J]. 计算机学报, 2002, 25(7): 759~766
- 8 李明, 黄文涛, 刘智云. 关于决策表约简的 CEBARKNC 算法改进 [J]. 计算机应用, 2006, 26(4)
- 9 张文修, 吴伟志, 梁吉业, 李德玉. 粗糙集理论与方法 [M]. 北京: 科学出版社, 2001