

一种度量软件需求的功能点方法

文 静¹, 顾勋梅²

(1 淮阴师范学院 计算机科学与技术学院, 江苏 淮安 223300; 2 淮海工学院 计算机工程学院, 江苏 连云港 222005)

摘 要: 基于目前应用广泛的功能点分析方法, 从软件需求规约的角度提出了合理度量面向对象系统的一系列映射规则和度量规则, 解决了基于人工解释所导致的度量结果的不精确问题, 并通过实例说明了这些规则的使用方法, 具有较强的实用性。

关键词: 功能点分析; 软件规约; 面向对象系统; 映射规则; 度量规则

中图分类号: TP311

文献标识码: A

文章编号: 1000-7180(2011)10-0075-04

A Function Point Method for Software Requirements

WEN Jing¹, GU Xun-mei²

(1 School of Computer Science and Technology, Huaiyin Normal University, Huaiyin 223300, China;

2 School of Computer Engineering, Huaihai Institute of Technology, Lianyungang 222005, China)

Abstract: Based on Function Point Analysis method used widely, a set of mapping rules and measurement rules are presented from software requirements specifications in order to estimate Object-Oriented systems appropriately. This improved method can solve the problem of imprecise measurement results because of human interpretation for calculating rules. Finally, a case study is introduced to explain how to use these rules to measure OO systems. It is improved that this method is applied well.

Key words: function point analysis; software requirements specification; object-oriented system; mapping rule; measurement rule

1 引言

软件项目的成本和工作量估算是以未来系统的规模为基础的, 所以, 在软件生命周期的早期阶段能够准确地度量出软件的规模显得尤为重要。功能规模度量(Functional Size Measurement, FSM)是从用户和功能的角度通过量化用户的功能需求来度量软件规模的一种方法, 适用于早期的规模度量^[1]。目前, 应用最广泛的 FSM 方法是国际功能点用户组织(IFPUG)提出的功能点分析(Function Point Analysis, FPA), 简称 IFPUG FPA^[1]。

由于标准的 FSM 方法是基于人类的解释, 导致了度量结果的不精确性, 因此, 本文基于面向对象方法的需求规格说明, 通过分析和改进 IFPUG

FPA 方法系统地定义了符合 FSM 标准的软件功能规模的度量过程, 并通过实例加以应用, 拓宽了 FPA 在软件生命周期内的适用阶段。

2 功能点分析

功能点分析(FPA)^[1]是在 20 世纪 70 年代初期, IBM 委托 Allan Albrecht 工程师和他的同事为解决 LOC 度量所产生的问题和局限性而研究发布的, 随后被国际功能点用户协会(the International Function Point Users' Group, IFPUG)提出的 IFPUG 方法继承, 在国际软件行业范围内得到推崇和一致接受, 并从单纯的规模度量发展到倾向于软件工程整个生命周期中的应用。2010 年, IFPUG 发布了 4.3 版本, 简称 IFPUG FPA 4.3。

收稿日期: 2011-05-13; 修回日期: 2011-06-30

基金项目: 江苏省高校自然科学研究项目(09KJD520002)

(1) 功能点分析的要素

功能点分析使用软件所提供的功能的度量作为规范值,它是基于软件信息领域的可计算的度量及软件复杂性的评估而导出的,具体的框架如图 1 所示.它将软件的功能归结为 5 个基本的功能要素:内部逻辑文件(Internal Logical File, ILF)、外部接口文件(External Interface File, EIF)、外部输入(External Input, EI)、外部输出(External Output, EO)和外部查询(External Query, EQ).有关这些功能要素的定义参见文献[1].

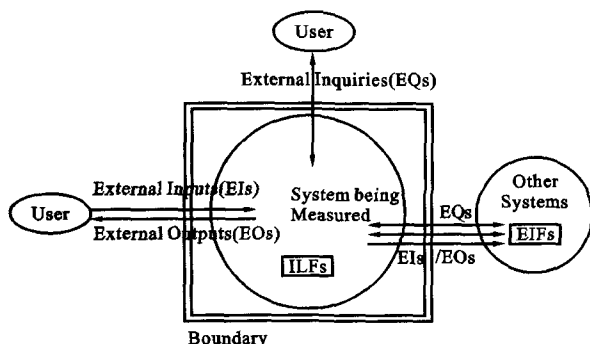


图 1 功能点分析的结构框架

这 5 个基本要素又可分为两类:数据功能(Data Function)和事务功能(Transactional Function).事务功能包括 EI、EO、EQ,数据功能包括 ILF 和 EIF.

(2) 功能要素的复杂度等级

每个功能要素的复杂度等级可以划分为“低”(low)、“平均”(average)和“高”(high).事务功能复杂度等级是由数据元素类型(Data Element Type, DET)和引用文件类型(File Type Referenced, FTR)决定的;数据功能复杂度等级是由 DET 和记录元素类型(Record Element Type, RET)决定^[1-2]. DET、RET 和 FTR 的定义如下:

DET 是用户可识别的无递归、不重复的信息单元,可以认为是一个数据元素、一个变量或一个字段. DET 是动态的,而非静态的,可以读自于文件,或由 FTR 的数据单元创建.

RET 是指在 ILF 或 EIF 中,用户可识别的数据集的子集,可以通过检查数据中的各种逻辑分组来识别他们,他们通常表现为一种父子关系.

FTR 是指在一个事务过程中,所引用到的各种文件,可以是 ILF,也可以是 EIF.

(3) 功能点计算的过程

FPA 是用于计算应用程序型(applications)、开发型项目(development projects)和升级型项目(enhancement projects)规模的.估算功能规模的过程

为:(1)确定功能点计算的类型;(2)识别计算范围和所要度量的应用程序边界;(3)识别和估算系统的各种功能要素及数量;(4)根据复杂度和加权值计算出未调整过的功能点数量 UFP(Unadjusted Function Point);(5)确定 14 个功能点校正因子,并计算出校正后的功能点数量^[1-2].

3 面向对象方法的开发过程

面向对象方法是一个自动的软件生产方法.该方法使用形式化语言来描述面向对象系统的规格说明,也可使用 UML 建模语言描述.图 2 描述了面向对象方法的软件生产过程,主要由 3 种高级概念模型(需求模型、概念模型、执行模型)组成^[3-4].

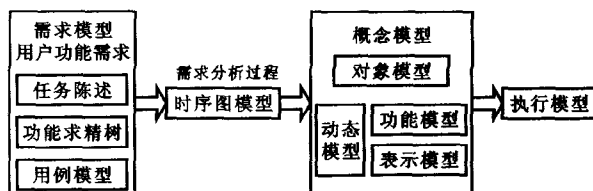


图 2 面向对象方法的软件生产过程

需求模型包括一组技术,如任务声明、功能求精树和用例模型等^[3].概念模型可以捕获系统功能需求中静态和动态的特性.面向对象方法可以将概念模型分为三个不同侧面的、相互补充的视图,分别为对象模型、动态模型和功能模型.执行模型可实现问题空间到解空间的转换,所以面向对象软件应用系统可在不同平台上系统地、自动地产生^[3-4].

3.1 需求模型

需求模型合并了一组互补技术来捕获系统需要的功能特性.需求建模所需要的技术如下:

(1)任务陈述:用 1~2 个简短的句子来描述软件系统的目的.通过任务陈述能清楚地知道系统的主要职责^[3].

(2)功能求精树:根据系统的不同事务域或不同事务目标,在不同的抽象层次将系统与外部的交互分割成不同的部分,并将这些部分用树型结构组织起来^[3].功能求精树主要是用来确定系统的用例.树根代表的是整个系统的功能(即任务声明),树的叶子节点代表的是系统的基本功能,而树的中间节点则代表部分基本功能的集合.

(3)用例模型:是从用户的角度来对系统功能需求进行建模^[3].功能求精树的叶子节点可以看成基本用例,当然还有一些次用例和功能求精树间接对应.另外,在实际应用中要密切重视用例之间的 INCLUDE 和 EXTEND 关系.

3.2 需求分析过程

需求分析过程(Requirements Analysis Process, RAP)是以用例模型描述用户需求为基础的,目的是确保需求规格说明和软件开发过程后续阶段的一致性^[3].在需求分析过程中,时序图用于描述对象之间的动态协作关系,每个用例至少对应一个时序图.时序图采用 UML 提供的标记以及并入消息的一些扩展符号,目的是为了把对象的交互划分成不同的种类,这样以后就能识别出面向对象概念模式的不同元素.

在用例图层面,主要任务是寻找直接参与建立时序图的角色.当用例接收到一个激活动作时,系统的内部组件就会产生一系列交互作为响应,这些内部交互在时序图中被表示成消息.根据消息不同特征,使用 UML 构造型^[4]可将它们分为以下 4 种:

(1) <<signal>>表示角色(外部实体类型)和系统之间传递的消息,其唯一的特点就是有方向性,具有两种类型的值:①Input,起点是一个角色类,接收者是边界类系统;②Output,起点是边界类系统,接收者是一个角色类;

(2) <<service>>表示更新对象状态的消息.如果一个对象被创建、删除或修改,那么 new/destroy/update 的操作就会被使用;

(3) <<query>>表示询问对象状态的消息;

(4) <<connect>>表示交互过程中在参与者对象间建立构造关系的消息.

另一方面,INCLUDE 关系使得基用例可以明确包含被包含用例,在基本时序图中通过消息调用来描述.而 EXTEND 关系只是用于扩展基用例.

4 基于面向对象方法需求模型的功能点度量规则

在面向对象系统中,用户功能需求(FUR)是通过需求模型获取的,功能规模数量是从需求规约说明中得到.因此,要度量的实体是由面向对象方法的需求规约组成.由于度量规则是基于 IFPUG FPA,所以度量过程必须符合 FPA 标准度量方法.

4.1 基于需求模型的功能点映射规则

映射规则主要用于识别需求模型中的元素, FPA 基本概念和需求模型元素之间的映射关系如表 1 所示^[3-6].

从表 1 中可以看出, EQ 和 EO 对应于需求模型中类型相同的消息,但是两者还是存在一定的区别: EO 必须包含数学推理或导出新的数据,所以,当且

仅当有其他消息提示查询有结果返回时 EO 才对应于 <<query>> 消息.此外, <<service/new>>、<<service/destroy>>、<<service/update>> 消息是用于维护目标对象的信息,也可以看成是 EI;同样,当 <<connect>> 消息是建立源与目标对象之间关系时,也可以看成是 EI.

表 1 FPA 基本概念和需求模型的映射关系

FPA 基本概念	需求模型
度量范围	用例模型
用户(user)	参与者
边界(boundary)	参与者和用例图之间的划分界限
内部逻辑文件(ILF)	类的内部实体
外部接口文件(EIF)	类的外部实体
外部输入(EI)	<<signal>> 构造型且具有输入值的消息
外部查询(EQ)	<<query>> 构造型且含有 <<signal>> 的消息
外部输出(EO)	<<query>> 构造型且含有 <<signal>> 的消息

4.2 基于需求模型的功能点计算规则

度量阶段的主要任务就是通过计算得到一个系统的功能规模值.根据 FPA 和需求模型概念之间的映射,基于需求模型的度量规则如下^[3-6]:

规则 1 类的每个实体看成是 1 RET;同时,每个外部实体也可看成是 1 RET.

规则 2 定义消息的实体类看成 1 FTR;一个消息引用的独立的实体类也应看成是 1 FTR;在用例前置条件中引用的独立实体类也是 1 FTR.

规则 3 一个实体的常量所引用的独立实体类是 1 FTR.

由于在需求模型中不能明确知道实体种 DET 的数量,因此,这里没有明确规定计算 DET 的规则.但可以根据实际系统需求进行约定,一般约定: EIF 和 ILF 给予 Low 复杂度等级,而 EI、EO 和 EQ 使用复杂度等级表中的中间列,然后根据 FTR 数量决定是哪个等级,最后得到相应的功能点数量.

规则 4 每个时序图的功能规模是图中涉及的所有 EI、EO 和 EQ 的功能点总和.

规则 5 每个用例的功能规模就是其时序图的功能规模.

但是由于用例之间通常存在两种关系(include 和 extend),因此这里附加两条关于基用例的计算

规则如下:

规则 6 一个用例的功能规模等同于基用例和扩展用例的功能规模总和;而包含其他用例的基用例的功能规模等同于被包含的用例和用例本身的功能规模总和。

规则 7 事务功能点就是用例的功能规模总和;数据功能点就是 ILF 和 EIF 的功能规模总和;系统的功能规模即为事务功能和数据功能之和。

5 实例研究

本节以我们曾开发的设备故障管理系统为例来说明以上规则的使用. 该系统的主要任务是自动管理单位设备所发生的故障,也包括设备的管理、费用的管理以及人员的管理等. 该系统主要包含 4 个功能组:故障管理、设备管理、费用管理和人员管理。

下面以设备故障管理功能组为例,说明度量的整个过程。

步骤 1:识别软件边界. 软件边界介于用例和参与者之间,因此故障管理模块的用例图如图 3 所示. 根据规则 1,图中的系统管理员、值班人员和用户参与者都可以看成是用户(user),再根据规则 2,可以识别出该软件系统的边界。

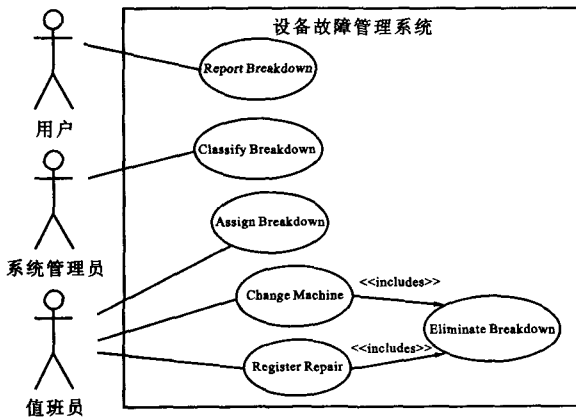


图 3 设备故障管理用例图

步骤 2:识别数据功能. 从功能求精树中获取的主要用例有:故障上报(Report Breakdown)、故障责任落实 (Assign Breakdown)、登记维修 (Register Repair)和置换新设备 (Change Machine);包含的用例有消除故障(Eliminate Breakdown). 根据各用例图,所识别出的 ILF 有:设备、系统管理员、值班人员、用户、故障设备;EIF 有代理值班人员。

步骤 3:识别事务功能. 这里仅以“登记维修 (Register Repair)”为例说明事务功能识别的过程. 登记维修对应的时序图如图 4 所示。

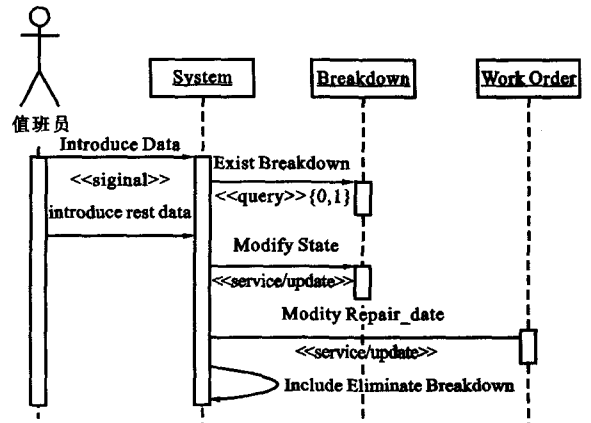


图 4 “登记维修”时序图

在该场景中,值班人员首先向系统发送故障检测信息,然后系统检验故障是否存在,如果存在,值班人员就导入其他信息,接着修改故障对象的状态和修理时间,最后在修理结束时调用“Eliminate Breakdown”次用例. 根据 4.1 节给出的映射规则,识别出 4 个 EI: Introduce Data、Introduce rest data、Modify State 和 Modity Repair date; 1 个 EQ: Exist Breakdown。

步骤 4:计算数据功能点. 根据上述规则,一个实体类就是一个 RET,但是在需求分析阶段不能确定 DET 的数量,因此对于每个 RET 分配 Low 复杂度等级. 因此,数据功能点为 $5 \times 7 + 5 = 40$ 。

步骤 5:计算事务功能点. 根据规则 1~规则 7,“登记维修”时序图共有 5 个 message,每个消息均操纵一个实体,即对应于 Low 的复杂度等级,因此,该时序图的事务功能点数量为 $3 \times 4 + 3 = 15$ 。

步骤 6:计算系统总的功能点数量,即数据功能和事务功能的功能规模总和。

6 结束语

本文介绍了一种基于需求模型度量面向对象系统的一种方法,其度量过程与 IFPUG FPA 方法很相似. 该方法避免了人工解释 FPA 规则的不精确性,能够在软件开发的早期即可获得软件的规模,减少了软件开发的風險,为软件项目开发的组织者提供决策支持,为企业提高软件项目的成功率奠定良好的基础。

然而,度量的精确性仍然值得探讨,如何提高度量结果的精确性以及如何实现自动化度量是下一步研究的方向。

(下转第 83 页)

5 结束语

MOS 位置的确定是一个综合决策问题,需要对其进行综合考虑制定最优策略.本文对影响 MOS 位置的主要因素进行分析,建立了一种 MOS 位置评价结构模型.为了避免出现模糊一致矩阵不一致的情况,利用 IFAHP 来确定评价指标的权重,并提出了一种新的量化方法构建优先判断矩阵.

本文的主旨在于说明 IFAHP 应用于 MOS 位置确定的可行性,因此 MOS 位置评价体系结构有些粗糙,而且在最后评价实例中,数据选取比较简单.如何完善 MOS 的评价体系,选取复杂的数据进行验证以及如何利用指标的权重进行 MOS 位置的确定是下一步作者的研究目标.

参考文献:

- [1] 杨文山,宋显宏,王驰,等.战时机场道面抢修技术[J]. 机场工程,2004(1):1-4.
- [2] 李明锋.机场最小起降带的确定方法分析[J]. 机场工程,2002(3):7-13.

- [3] 许巍.模糊 AHP 法在机场最小起降带优选评价中的应用[J]. 航空计算技术,2002,32(3):12-14.
- [4] 许巍.机场最小起降带模糊优选理论模型[J]. 交通运输工程学报,2005,5(1):57-60.
- [5] 王家顺,王田苗,胡耀光,等.一种基于模糊层次分析法的供应商评价模型的研究[J]. 微电子学与计算机,2001,18(2):59-64.
- [6] 腾继东,项梦杰,李苏,等.改进模糊层次分析法确定地址灾害危险性评价指标权重的研究[J]. 安徽农业科学,2008,36(1):22-23.
- [7] 卢震.基于模糊层次分析法的军队专业技术人才考评[J]. 北京航空航天大学学报:社会科学版,2010,23(1):68-74.
- [8] 周艳美,李伟华.改进模糊层次分析法及其对任务方案的评价[J]. 计算机工程与应用,2008,44(5):212-214.
- [9] 张吉军.模糊一致判断矩阵 3 种排序方法的比较研究[J]. 系统工程与电子技术,2003,25(11):1370-1372.

作者简介:

邢建春 男,(1964-),教授,博士生导师.研究方向为国防工程智能化、计算机测控、人工智能等.

(上接第 78 页)

参考文献:

- [1] IFPUG. ISO/IEC 20926: Software engineering - IFPUG 4.3 Unadjusted FSM Method-Counting Practices Manual [M]. 2010.
- [2] 李帆,林立新,曹亚波.软件工程项目管理——功能点分析方法与实践[M]. 北京:清华大学出版社,2005.
- [3] Insfran E. A requirements engineering approach for object-oriented conceptual modeling[D]. Valencia: Valencia Polytechnic University, 2003.
- [4] Object Management Group - OMG. Unified Modeling Language-UML v 1.5[M]. 2003.
- [5] Abrahao S, Pastor O. Estimating the applications func-

tional size from object-oriented conceptual models [C]// Proceedings of IFPUG'01. Las Vegas, USA: IEEE, 2001.

- [6] Juan J. Cuadrado Gallego, Fernando Machado-Piriz, et al. On the conversion between IFPUG and COSMIC software functional size units: A theoretical and empirical study[J]. The Journal of Systems and Software, 2008(81):661-672.

作者简介:

文静 女,(1978-),硕士,讲师.研究方向软件方法学、软件质量保证.