框架设计通用化方法研究*)

于显平　余建桥
(西南农业大学信息学院　重庆 400716)

摘　要　本文系统地讨论了框架设计的基本概念、原理，提出了框架设计的通用化方法，即应用建立高层逻辑模型、视图合并和回归验证方法实现框架设计，并应用该方法实现了办公自动化系统框架设计和实例化。

关键词　框架，视图，办公自动化

Study on the Generalization of Frameworks Designing

YU Xian-Ping　YU Jian-Qiao
(Information College, Southwest Agricultural University, Chongqing 400716)

Abstract　This paper explores the concept and principle of frameworks designing and proposes an approach to the generalization of keyframe designing, namely, designing frameworks by building high-level logic model, view-combination and recurved certification. Using the approach, this paper also achieves OA system frameworks designing and instantiation.

Keywords　Framework, Viewpoint, OA

现代软件工程越来越强调软件的质量保证和软件的可重用性。软件的可重用性从代码可重用、构件可重用发展到设计的可重用。框架是指在一个特定的领域中的一组相互协作的类，它收集了常用于该应用领域的设计决策，规定了应用系统的总体结构，定义了类和对象的划分，定义了其关键责任，定义了类和对象如何合作，还定义了控制线索。一个定义良好的框架对于解决该应用领域中的软件开发和设计具有较高的复用价值。

框架设计是属于逻辑上较高层次的软件设计范畴。20世纪90年代以来，对框架设计理论做了大量研究，提出了UML-F框架表示方法、模板-钩点框架构建理论等，但仍没有一个非常成熟的通用化方法进行框架设计，框架的实例化及其验证也难以实现。

框架设计的切入点是系统分析和设计，通过系统分析获得高层逻辑模型、实现视图归并、生成模板-钩点模型、消去热点联系并回归验证完成框架设计。最后通过办公自动化系统框架设计实例验证并总结了框架设计的通用化理论与方法。

1　框架基础知识

框架是一个系统全部或部分的可复用设计，通常由一组抽象类和类之间的协作组成[1]，因而框架可以采用UML统一建模语言来描述类及其相互间的关系，并且可以应用其扩展机制表示框架的扩展点。框架分为两种：面向对象框架(Object-Oriented Framework，简称OOF)和基于构件的框架(Component based framework，简称CBF)。OOF的复用是通过对框架中的抽象类进行特殊化的方式来定义框架行为的，每一个抽象类派生出一个子类，并在子类中给定所有纯虚方法的具体实现，复用这些子类来开发特定应用系统；CBF的复用将基于继承的面向对象框架通过用构件接口中方法的调用来替代对象类中方法的重载，转换成为基于构件的框架。是由相互协作的构件组成，并通过对构件接口的扩展来实现系统[2]。

钩点(Hook-spot)也称为凝固点，指系统中相对固定不变的部分，即系统核心功能部分；热点(Hot-spot)是指应用系统中灵活变化的部分[3]。热点与钩点的有效区分，模板钩点模型的创建，对于框架设计具有重要意义。

2　框架设计通用化方法

2.1　系统逻辑模型导出系统架构

框架集中了某领域软件设计的共同性质，因而需要对该领域充分认识，并以恰当的工具来表示。面向对象建模技术可以首先完成该领域软件的高层逻辑模型，再结合几个具体应用，抽象出该领域软件的共性，大致形成系统架构。

本研究主要以办公自动化系统软件框架设计为例进行讨论。图1是一个办公自动化系统的高层逻辑模型，简单地定义了主要的功能需求，同时忽略了一些非共性的功能。

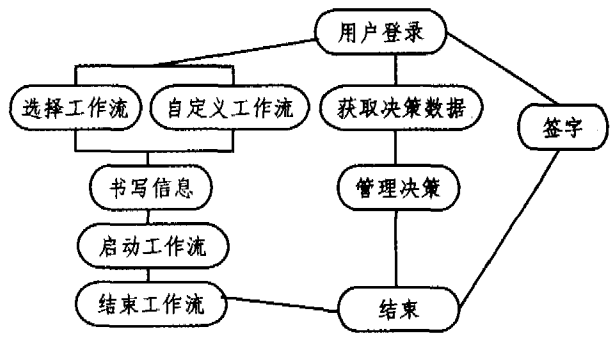

图1　办公自动化系统逻辑模型

*)重庆市科委项目(20038035)资助。于显平　硕士，主要研究方向是人工智能、软件工程等。余建桥　博士，教授，主要研究方向是人工智能、数据库系统、软件工程等。

2.2 视图归并形成模板钩点模型

高层逻辑模型对于系统的认识和了解是从较高层面的，但要具体地分析软件的功能，特别是软件的基本功能和其功能扩展，就必须进行细化。其中对视图的分析和合并将是解决问题的主要办法。

视图是从不同的角度描述系统的不同方面，如 UML 语言通过用例图、类图、对象图、状态图等多种图例实现了系统的完整说明，使得对系统的理解更加准确。框架设计就是通过对各分解视图进行合并，特别是对不同应用系统的视图比较合并，得出其凝固点和热点，形成模板钩点模型。

视图 1 和视图 2 分别是高校办公自动化系统和企业办公自动化系统关于工作流的视图。在生成工作流时，首先需要选择员工或部门；然后由于工作流的类型有很多，例如可以是普通文档，也可以是图形图像、图表，甚至是企业生产过程零部件的报表等；在保存时需要选择是保存到模板（对于一些固定生产流程报表等）或者临时工作流（如临时产生的请示、批示等）。所以形成了如图 2 所示的工作流视图模型。

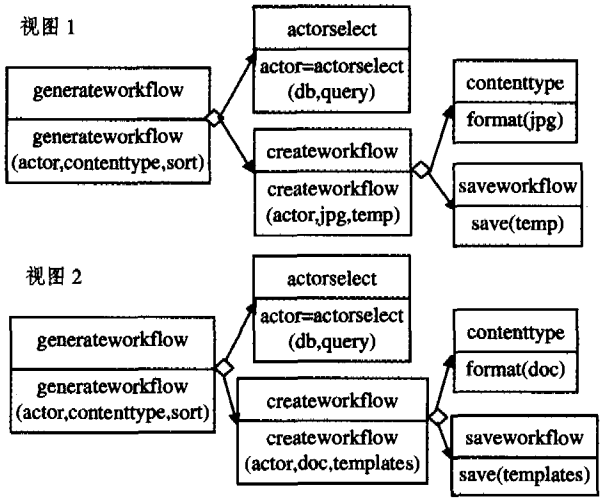

图 2 工作流视图模型

由于不同的应用系统、不同的出发点，不同的人都可能产生不同的视图，必须对它们进行分析合并，才能找出其中的共性。视图合并的基本原则是：①同一概念内容的名称必须一致；②各概念相互独立且不相互包含；③概念属性及其类型必须一致；④不允许循环继承。图 2 中关于工作流视图实际上已经进行了合并。

接下来产生模板钩点模型。可以依据如下规则[4]：

(1)所有视图中具有相同名称的对象直接映射到模板钩点模型中的类；

(2)在所有视图中具有相同名称、参数和内容的方法直接映射到模板钩点模型中的方法；

(3)在所有视图中具有相同名称、不同参数和内容的方法形成模板钩点模型中具有相同名称但未定义参数和内容的钩点方法；

(4)在所有视图中具有相同名称、相同参数和不同内容的方法形成模板钩点模型中具有相同名称、相同参数但未定义内容的钩点方法；

(5)所有拥有钩点方法的方法仍被定义为模板方法，但分别拥有模板钩点方法的类之间存在着热点关系，即它们是变化因数必须进行处理；

(6)其它所有联系均继续保留在模板钩点模型中。

通过前述规则，就可以导出系统每一局部的模板钩点模型。图 3 是工作流的模板钩点模型。

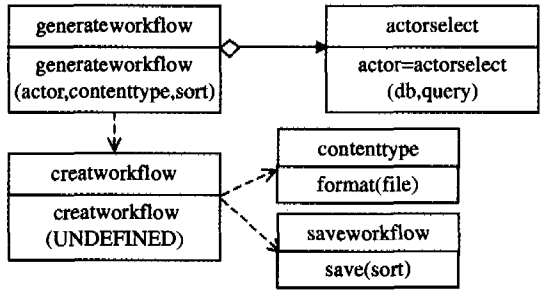

图 3 工作流模板钩点模型

其中，虚线表示的是热点联系，即箭头所指的类或方法是变化的。

2.3 消去热点联系并回归验证，形成框架设计

框架设计的一个基本原则就是尽量减少设计复杂度，对变化的类或对象进行尽可能的抽象，再利用扩展机制实现其热点。上述模板钩点模型要上升到框架，必须消去热点联系，才能识别抽象类、子类，对原本属于抽象类的抽象方法或具体方法才能有效地获得。在普通情况下，框架设计者采用的是热点卡（Hot-spot cards）的方式，应用设计元模式实现它。但是元模式应用比较复杂，难以真正采用。本文拟采用消去热点联系并回归验证的方法实现框架设计。

首先，分析模板钩点方法是否属于同一个 ，如果是同一个类，而且钩点方法是固定不变的，就可以合并为一个类，钩点方法所在的类撤销，合并到上一级类中作为一个新的方法。如示例中 contenttype 以及 saveworkflow 都属于同一个类 createworkflow，另两个类的方法是固定不变的，则合并到类 createworkflow 中成为两个方法。

模板钩点方法不属于同一个类时，将模板方法所在的类形成抽象类，钩点方法所在的类定义为子类，利用继承的方式实现其实例化。

对已形成的框架设计，利用未合并的视图进行回归验证，找出其中类或方法的归并可能导致的错误，重新应用前述规则，进行合并。当然验证必须是在对系统充分熟悉的基础上进行的并有相当的难度。图 4 就是最后的工作流的框架模型。

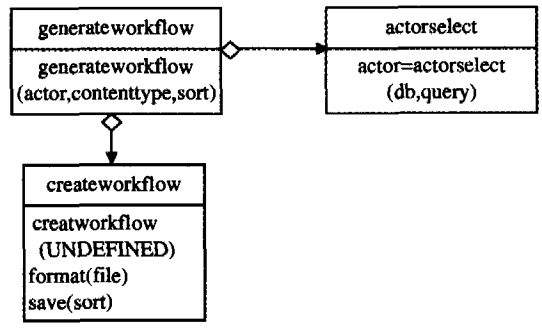

图 4 工作流框架模型

3 框架实例验证

通过应用上述通用化方法，设计了办公自动化系统框架，图 5 是办公自动化系统框架模型。并最终实例化了该办公自动化系统框架，实现了企业办公自动化设计，并将企业日常生产管理工作纳入了办公自动化，极大地扩展了办公自动化的

应用范畴。

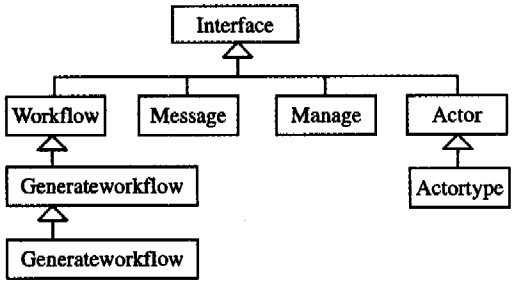

图 5 办公自动化系统框架模型

相关工作 框架设计方法非常复杂,对其程序化一直是我们研究的重点。回归分析设计框架完成了框架的准确构建,但其它的附加内容非常多。例如,热点的形式化描述,其实例化的具体解决办法,设计模式的有效应用,框架的扩展点的处理等都是我们需要进一步研究的内容。

结束语 通过系统研究框架设计通用化方法,总结了一条切实可行的框架设计方法,特别是提出了通过回归验证消去热点联系的新思路,并通过设计办公自动化系统框架进行验证。事实证明是可行的而且高效。在实例化办公自动化系统中,采用了面向对象框架设计技术和面向构件技术相结合的办法,取得了良好效果。当然,对本文提出的方法只是在几个实例中进行了验证,并没有进行理论证明。尤其是设计模式、UML-F 语言等多种理论知识和实践方法应进一步应用于本方法,使从框架设计到实例化的通用化方法更加完善。

参 考 文 献

1 Fontoura M, Pree W, Rumpe B. UML-F: A Modeling Language for Object-Oriented Frameworks. ECOOP' 2000, LNCS 1850, Sringer-Verlag, 2000. 68~32
2 胡文蕙,赵文,张世坤,等. 基于构件技术的应用框架元模型的研究. 软件学报,2004 ,15(1)
3 Pree W. Hot-Spot-Driven Framework Development
4 Fontoura M, Crespo S, Lucena C J. Using viewpoints to derive object-oriented frameworks: a case study in the web-based education domain. Journal of Systems and Software, 2000, 54(3): 239~257
5 Olivereira T C, Alencar P S C. Software Process Representation and Analysis for Framework instantiation. IEEE. Transactions on Software Engineering, 2004, 30(3)
6 Johnson RE. Frameworks=(Components+Patterns). Communications of the ACM 1997, 40(10): 39~42
7 Froehlich G, Hoover H, Liu L, et al. Hooking into Object-Oriented Application Frameworks. ICSE'97, IEEE Press, 1997. 491~501
8 Lewis T, et al. Object-Oriented Application Frameworks. Greenwich, CT: Manning Publications/Prentice Hall, 1995
9 Goldberg A, Rubin K. Succeeding with Objects-Decision Frameworks for Project Management. Reading, Massachusetts: Addison-Wesley, 1995
10 Pree W. Object-Oriented Design Patterns and Hot Spot Cards. IEEE Intel. Conf. on the Engineering of Complex Computer Systems (ICECCS'97), Como, Italy, 1997

(上接第 265 页)

级和 3 级的 18 个过程域),我们搜集了所有评估参与者(共 5 人)的问卷数据。通过分析得到了问卷统计表、每个目标的满意度以及每个过程域的满意度等。

图 4 显示了过程域"需求管理"中每个实践(包括 5 个特定实践和 10 个共性实践)在该项目中的实施程度。由图可知,该过程域中所有实践的实施程度都是大部分实施或全部实施,满足所有共性目标和特定目标,因此可认为该过程域达到满意。

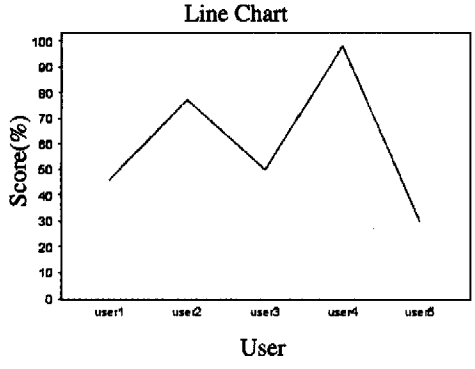

图 5 "确定需求接口"实践的答案分布

图 5 显示了对于过程域"需求开发"中的特定实践"确定接口需求",所有答卷者给出的答案分布。由图可知,不同的答卷者给出的答案差距较大(方差值较大),说明在该项目级实践的实施状况上存在意见分歧。有必要对该实践进行再分析和评估。

结论 MSPSA 以 CMMI 为参考模型,支持 SCAMPI 的 3 个阶段,遵循小型过程评估的思想,具有耗费资源少、评估周期短的特点,可以很好地满足中小型组织的软件过程改进,以及大型组织快速了解内部过程能力状态的需求。MSPSA 主要的缺陷在于:①评估准确度不高。由于缺少现场取证、验证阶段,评估结果受答题者主观因素影响很大。②评估的覆盖面有限。评估考察的是组织的软件过程与参考模型相比较的实施程度。如果组织采用其他的途径达到模型中软件过程的目标,MSPSA 则无法反映。

下一步的工作集中于:完善评分算法,通过赋予不同的评估参与者以不同的权值,提高评估结果的合理性;通过赋予问题间的关联,尽量减少答题者主观因素的影响。

参 考 文 献

1 Carnegie Mellon Software Engineering Institute. CMMI-SM for Systems Engineering/Software Engineering, Version 1.02, Staged Representation. 2000
2 Carnegie Mellon Software Engineering Institute. Standard CMMI Appraisal Method for Process Improvement (SCAMPI), Version 1.1: Method Definition Document. 2001
3 Carnegie Mellon Software Engineering Institute. Appraisal Requirements for CMMI-SM, Version 1.1. 2001
4 Carnegie Mellon Software Engineering Institute. Upgrading from SW-CMM to CMMI. 2003
5 Wiegers K E, Sturzenberger D C. A Modular Software Process Mini-Assessment Method. IEEE, 2000
6 Humphrey W S. Managing the Software Process. 北京:清华大学出版社,2002
7 罗运模,谢志敏,等. CMMI 软件过程改进与评估. 北京:电子工业出版社,2004
8 Ahern D M, Clouse A, Turner R. CMMI 精粹——集成化过程改进实用导论. 北京:机械工业出版社,2002
9 卡耐基梅隆大学软件工程研究所. 能力成熟度模型(CMM):软件过程改进指南. 北京:电子工业出版社,2001