文章编号：1000－5471(2007)04－0082－05

一种基于逻辑约束的软件设计过程验证方法及其工程实践①

何俊梅， 邹显春

西南大学 计算机与信息科学学院，重庆 400715

摘要：如何在软件设计过程中对软件需求加以正确描述和验证一直是软件开发过程中困扰开发者的重要问题之一. 在传统 UML 面向对象的分析与设计方法的基础上，提出一种利用 RSL 对系统进行逻辑约束，在开发设计过程中不断进行系统逻辑规约验证的有效反方法. 该方法提高了系统的稳定性与可靠性. 我们还将该方法用于开发了一个教学管理系统以验证该方法的系统设计能力.

关 键 词：统一建模语言；逻辑约束；教学管理平台；RAISE 规范语言

中图分类号：TP311.5 **文献标识码**：A

随着软件技术的不断发展，开发者和用户对于软件质量提出了更高的要求. 为此，软件开发者试图从技术、管理等各层面控制软件开发过程，提高软件产品的针对性和可靠性，保证软件对于用户的使用价值.

在现有的技术中，开发者用于保证软件开发质量的主要手段可以归纳为以下 5 种：

(1) 采用更加科学实用的软件开发过程模型.

(2) 使用高效的软件设计和开发方法.

(3) 在开发过程中使用基于严格数理逻辑的形式化方法指导软件开发过程.

(4) 在软件开发中，使用面向不同软件抽象层次的软件测试策略.

(5) 在软件开发过程中应用适当的项目管理技术.

在实践中，通过上述方法的综合应用，有效的提高了软件产品开发的效率和成功率，保证软件质量. 在众多软件质量保证技术中，软件开发过程中的验证和质量保证方法相对于软件测试，软件项目管理等方面的技术而言一直存在诸多技术难点. 而造成这些难点的主要原因在于软件开发过程中夹杂了大量的不确定因素，传统的方法往往更多的寄希望于软件开发者个人能力的发挥.

1 UML 的形式化约束限制描述方法

UML 使用图形方式刻画软件系统的实现细节，为软件设计带来了诸多方便，但是由于图形工具的局限性，使得 UML 在刻画软件系统的潜在限制开发规约方面缺乏完整有效的途径和手段[1,2]. 而良好的操作语义对开发和测试都起着至关重要的作用，从软件的角度来看，与软件设计相关的类定义操作语义主要包括三个方面：

① 收稿日期：2006－12－14

基金项目：西南大学教改项目 (2006JY44).

作者简介：何俊梅(1967－)，女，重庆人，讲师，主要从事数据库技术、软件工程的研究.

前置条件：前置条件是当操作被执行之前应该满足的条件. 它通常是根据以下两种情形来描述的；一种是包含操作的对象的属性，另一种是请求执行这一操作的消息中所包含的实际参数的属性.

后置条件：后置条件是当操作被执行结束后所满足的条件. 它通常根据以下几点来描述：

(1) 包含操作的对象的属性变化情况；

(2) 请求执行操作的消息中所包含实际参数的属性的变化情况；

(3) 根据任意返回值来决定；

(4) 根据可能出现的异常来决定；

在类中，所有操作说明的集合对类实例的行为做了部分的描述. 行为很难单独地从操作中推断出来，所以行为通常是经过状态转换的更抽象的表现形式. 行为被形象的定义为实例的一系列状态，而且还描述了各种操作怎样影响状态之间的转换.

相关性条件：相关性条件是后置条件的一种特例，主要指操作之间或操作与属性之间的相互关联条件，其基本规则是一个操作的结果不能够对其它操作产生不良影响. 因为其在测试过程中的重要作用而被单独定义.

在 RSL[3,4] 中前置条件包括：

(1) 操作中常量和变量的类型和属性；

(2) 操作的前提(一般用关键字 pre 描述)

后置条件包括：

(1) 操作的执行方法

(2) 操作的结果限制(一般用关键字 post 描述)

相关性条件主要是指操作与操作之间、属性之间、操作与属性之间的相关性限制，描述为：[operation1_operation2]

[operation_attribute]

[attribute_operation]

[define]

为了进一步刻画软件的开发规约，我们采用传统的形式化规约语言 RSL 对系统的 UML 设计图形加以扩充和限制[5,6]. 此项工作的目的主要在于增强对系统规约的描述能力，更加清晰的限定系统边界，这将为系统的可靠性和稳定性的提升带来很大的好处.

规则 1　使用前置条件和后置条件进行软件约束定义的算法如下：

在算法中，pre[　]是对应操作的前置条件集合，post[　]是对应操作的后置条件集合，limitcase 为一个系统约束集合，in 为可能的输入，exp 为预期的正确结果.

```
procedure condi_combin
begin
set pre={ };
set post={ };
pre=combin(conditions); //add the possible pre-condition combination
                        //into the pre by the table 1;
post=combin(conditions); //add the possible post-condition combination
                         //into the post by the table 2;
check-model= pre×post //make the set check-model as the Descartes Set
                      //of pre and post.;
if (the condition combination is nonsensical)
```

then delete the combination from the check-model set.

end

规则 2 通过 RSL 的相关性约束产生系统约束的输入集合为对应的属性值和约束中描述的操作行为，系统的预期结果为相关性约束中的相等或从属等结果关系描述.

值得注意的是，在从最初的 UML 类图转换到 RSL 的过程中，刻画软件的相关性条件比较少. 其主要原因在于 UML 是一种非形式化的软件描述方法，它的主要优点在于描述软件的结构关系，而在描述某些具体的内部相关性关系时存在着一些不可避免的缺陷，即使用 OCL 作为补充，也会因为 OCL 自身的局限性受到很大的限制. 另一方面绝大多数的软件设计者在使用 UML 时更注重使用 UML 的图形描述方法而忽略其限制条件.

2 综合实践学习系统需求分析

随着高校计算机基础课程的不断改革与深入，传统的课堂教学与实验教学方式已逐渐呈现出严重的不足，其中又以实验教学为甚. 我们知道，计算机基础课程的实践性很强，据资料显示，许多高校的实验教学学时均高于课堂教学学时，这充分地说明，若没有强有力的实践保证，难以全面提高课程的学习效果，为此，我们开发了综合实践学习系统，利用校园网使学生可以随时进入实验学习环节. 系统从人性化、个别化、渐进化角度出发，帮助学生解决在实验环节中的片面性与盲目性，同步或异步地回答学生提出的每一个问题，并结合教育心理学知识，帮助学生克服某些心理障碍，有效地指导学生完成实验，从而促进其学习兴趣的提高，自主学习能力的增强，从根本上弥补了以往限时限地的实验教学方式所产生的缺陷.

本系统是对课堂课程学习的一个有力补充，主要是帮助、指导学生完成各种难度不等、覆盖面不等的综合性实验题目. 通过本系统所提供的各种教学策略，尽可能地激励学生的创造意识和创新意识，促进学生对知识的主动建构.

本系统由上机实践子系统和辅导答疑子系统构成，系统的角色有学生和教师. 在系统的顶层，抽象确定两个用例，即“上机实践子系统”和“辅导答疑子系统”. 将“上机实践子系统”分解为“上机注册”、“选择项目”、“查询课程信息”等多个低层用例，“辅导答疑子系统”分解为“教师注册”、“在线辅导”和“查询学生信息”等多个低层用例.

这里以“上机实践子系统”为例，用对象类图来描述系统的静态结构特征和行为，给出对象类及其联系，如图 1 所示，图中老师通过回答学生的留言与“上机实践子系统”进行交互.

使用 RSL 描述语言，可以有效的描述各个类的形式化约束，以下为类 Student 的形式化约束定义：

SET =

class student

type

name : Set_name

password: Set_pass

droit = = empty | add(droitElem,Set)

droitElem : {teacher, student, manager.....}

value

getpassword: name→password

login: name×password→Bool

register : name×password×droit→Bool

axiom forall n, n_1, n_2: name; p_1, p_2: password; d_1, d_2: droit

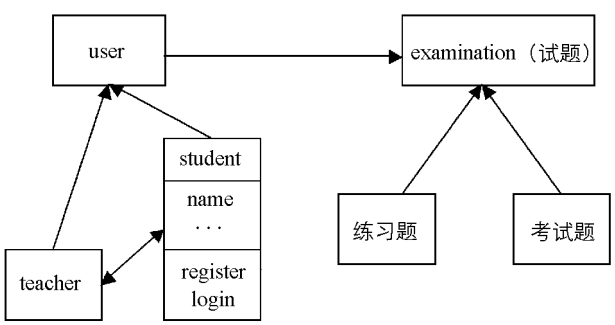

图 1 “上机实践子系统”对象类图

login(n_1, p_1)≡if (n_1∈Set_name) and (p_1=getpassword(n_1)) then true else false,

getpassword(n)≡if n∈Set_name then return p else false,

register(n_2, p_2, d_2)≡(Set_name=Set_name ∪ n_2) and funmap(n_2, p_2) and droitmap(n_2, d_2)

[only_password]

getpassword(n_1)≡getpassword(n_2)⇔$n_1=n_2$

[limit_name]

$\exists n$: $Set\_name|legth(n)>=6$

[limit_password]

$\exists p$: $Set\_pass|length(p)>=6$

end

由上述定义所得到对象 user 的约束条件见表 1.

表 1　对象 students 的语义约束分析

	前置条件	后置条件
getpassword	name : Set_name	Set_pass
login	name : Set_name ∧ password: Set_pass	if (n_1 ∈ Set_name) and (p_1 = getpassword (n_1)) then true else false,
register	droit:empty | add(droitElem, Set) ∧ name : Set_name ∧ password: Set_pass	

而对象的相关性约束包括：

[only_password]

getpassword(n_1)≡getpassword(n_2)⇔$n_1=n_2$

[limit_name]

$\exists n$: $Set\_name|legth(n)>=6$

[limit_password]

$\exists p$: $Set\_pass|length(p)>=6$

这些约束分别描述了对象 user 在使用过程中的各种限制和行为准则，是开发过程中必须遵守的. 同时对于系统约束的遵守也将有效地提高系统的可靠性和准确性. 同时可以采用类似的方法定义其它对象的数据约束条件.

而相关的系统约束将作为测试断言在软件代码实现的过程中嵌入到正常代码当中，作为实时的系统检验工具在系统运行时自动检查系统运行状态，并即时作出相关错误报告.

3　应用效果分析和结论

通过对于系统约束条件的提取和测试断言的嵌入，系统相对于过去的系统在运行稳定性和可靠性上有了很大的提高. 系统 4 小时可靠性测试的通过率从过去的 89%提高到现在的 99%，已经能够很好的满足用户对于该系统的需求，受到了广大师生的一直好评.

综合实践学习系统的开发有效地弥补了高校计算机基础课程实验资源(包括硬件资源与软件资源)严重不足的缺陷，更符合现代教育学的理论，可以充分发现与挖掘学生自身的潜力. 不过，任何教学都离不开老师的指导，因此，只有将本系统与课堂实验指导相结合，才能发挥其强大的作用.

参考文献：

[1]　练凯波. uml 建模语言[DB/EB]. http://www.kecourser.com/free/uml/page1/Java_UML.php, 2006－03－10.

[2]　林鄂华，席壮华. 从应用系统设计看 UML[DB/EB]. http://file.swufe.edu.cn/～e_learning/researchdoc/network/5.

doc,2006－03－05.

[3] C. George, A. E. Haxthausen, S. Hughes, R. Milne, et al. The RAISE Development Method[M]. Denmark: TERMA Elecktronik AS, 1999.

[4] C. George, A. E. Haxthausen, S. Hughes, R. Milne, et al. The RAISE Specification Language[M]. UK: Prentice Hall International Ltd, 1995.

[5] D. J. Andrews, J. F. Groote, C. A. Middelburg. Semantics of Specification Language[M]. Utrecht: Spring-Verlag, 1993.

[6] PRIESTLEY M. Practical Object－Oriented Design with uml[M]. 北京：清华大学出版社，2000.

[7] 周 玮，孙挺姝. 综合性学习网站的设计研究[J]. 中国电化教育，2005，(216)：80－83.

[8] 王继杰，陈声链，黄万华. 基于UML的综合教务管理系统的分析与设计[J]. 计算机时代，2004(11)：30－32.

A Software Development Process Validation Method Research and Practice Based on the Logic Restrict Condition Management

HE Jun-mei, ZOU Xian-chun

Faculty of Computer and Information Science, Southwest University, Chongqing 400715, China

Abstract: One of the most important problems of software development is the correct description and validation about the software requirements. Based on the traditional UML object-oriented methods of software analysis and design, the authors of this paper propose an efficient method to have logic restriction to the system with RSL, which carries on logic validation to the system development. It improves the stability and reliability of the developed system and has been applied to develop an instruction management system successfully.

Key words: UML; logic restriction; instruction management platform; RSL

责任编辑 张 枸