

# 需求变更自动化管理模型与实现<sup>①</sup>

郭新峰<sup>1</sup>, 马世龙<sup>2</sup>, 吕江花<sup>2</sup>, 武俊生<sup>1</sup>, 邵回祖<sup>1</sup>

<sup>1</sup>(山西大学 软件学院, 太原 030013)

<sup>2</sup>(北京航空航天大学 计算机学院, 北京 100191)

**摘要:** 需求变更不可控, 需求变更常对软件开发产生严重影响, 需求变更管理是控制影响的直接途径. 为易于理解沟通, 基于资源、组织与业务的需求模型(简称 ROB)用递归分解的方法分别从资源、组织与业务抽取需求, ROB 全局统一的树形结构便于形式化定义需求对象以实现在计算机中存储和管理; 在规范需求变更管理组织、流程基础上, 建立数据模型, 设计数据操纵语言定义操纵接口, 通过编辑修改对象模拟需求发生变更, 在完整性、一致性规则约束下, 自动完成由需求变更引发的完整操作模块. 实践中应用 ROB 需求管理软件进行需求变更管理验证这种方法的有效性、自动化智能化等特点.

**关键词:** 需求建模; 需求管理; 需求变更; 需求评审; 需求跟踪; 变更控制; CMMI

## Implementation of Automated Management Model for Requirement Change

GUO Xin-Feng<sup>1</sup>, MA Shi-Long<sup>2</sup>, LV Jiang-Hua<sup>2</sup>, WU Jun-Sheng<sup>1</sup>, SHAO Hui-Zu<sup>1</sup>

<sup>1</sup>(College of Software, Shanxi University, Taiyuan 030013, China)

<sup>2</sup>(Computer College, Beihang University, Beijing 100191, China)

**Abstract:** Changes of software requirement are uncontrollable. Requirement change and its chain reactions are the most influential risk factor for the project quality during any stages of software development. Automatical requirement management can minimize the negative impact of requirements change. Automaticity of requirement change management relies on formal level and degree of standardization. Based on ideas of recursion and decomposition, this paper analyzes the requirement analysis method, referred to ROB. ROB is closer to reality. ROB is easily understandable and acceptable. ROB is a requirements modeling approach based resource, organization and business. It makes an improvement on management processes and management organization, designs the storage model by formalizing the requirement as a tree. It achieves the automatical and intelligent requirement change management by modify the tree under certain rules and constraints. In practical applications, the effectiveness of our approach has been indicated with a software tool.

**Key words:** software requirement modeling; requirement management; requirement change; requirement review; requirement track; requirement change control; capability maturity model integration(CMMI)

软件依据需求, 需求对软件有重要影响. 需求发生变化的现象普遍存在是已被公认的事实. 需求变更不能得到有效管理, 常使软件开发处于失控、混乱状态. 需求变更进行科学管理, 对于保证软件质量起到至关重要的作用<sup>[1,2]</sup>. 需求变更管理任务包括: 1. 评估必要性、合理性, 决定是否实施需求变更; 2. 需求变更

能自动化程度较高的完成; 3. 可记录、存储需求变更, 以便必要时回溯恢复. 近年来, 需求变更管理逐渐引起人们重视, 相关研究取得了一定成果. 但面对软件的复杂性<sup>[3]</sup>, 缺少有效的需求变更管理模型及相应的实现方法, 需求变更没有得到科学的管理<sup>[4]</sup>.

本文第 1 节基于递归和分解的思想提出资源、组

① 基金项目: 国家自然科学基金(61300007); 软件开发环境国家重点实验室基金(SKLSDE-2012ZX-28, SKLSDE-2013ZX-11)

收稿时间: 2014-07-19; 收到修改稿时间: 2014-09-09

织与业务的需求建模方法(Requirements modeling of multi-viewpoints on Resource, Organization and Business 简称 ROB); 第 2 节在 CMMI 指导下设计了需求变更管理流程、组织和度量体系; 第 3 节建立了需求变更的管理模型; 第 4 节形式化定义了统一接口, 设计了对应的操纵语言, 通过对需求对象编辑修改, 实现需求变更的自动化管理; 第 5 节比较相关工作, 总结全文并提出进一步工作。

## 1 ROB 需求定义

随着软件应用领域增多<sup>[5]</sup>, 规模逐渐变大, 需求越来越复杂, 需求分析越来越难。需求在软件工程各阶段可能发生变更, 需求变更更引发设计、编码直到测试维护阶段一系列的反复和修改工作。需求变更管理的有效性依赖于自动化程度, 自动化管理建立在形式化基础上。文献[6-8]探讨获取、更新变化中的需求, 以形式化方式规范细化, 消除冗余提高抽取效率。但目前关于需求变更管理过程进行建模抽象、形式化规范并存储操纵等研究还不多见。为说明 ROB 需求变更管理模型, 首先给出 ROB 模型的结构和定义。

### 1.1 需求建模的工作模式

ROB 在规范工作模式(图 1)下进行建模活动, 包含四个层次: ①交互层, 需求分析人员与需求提供者沟通, 将软件作用到现实世界真实现象描述为自然语言为主的非结构形式; ②需求子模式层, 从交互层通过抽取、分类分解等过程, 分别以形式化方式建立树形结构的局部需求对象; ③需求模式层, 通过完整性、一致性检验并消除冲突与冗余, 从全局视角将需求对象整合形式化为树形结构整体需求对象, 即完整需求树; ④需求内模式层, 将完整需求树转化为数据模型

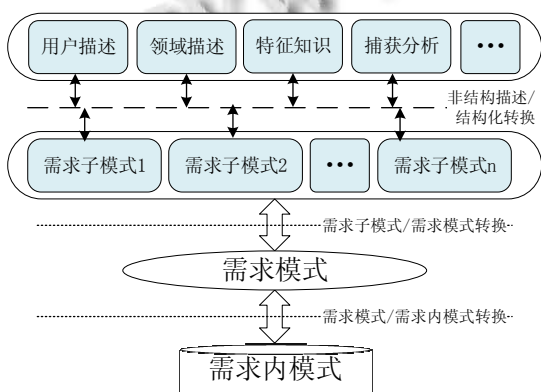


图 1 ROB 建模工作模式

(DOM 对象树模式、关系模式)在计算机中存储, 定义实现操纵语言完成需求数据的自动化处理。

### 1.2 ROB 体系结构

目前需求模型相关研究具有代表性的如: 面向本体的需求分析(OORA)、面向目标的需求模型、面向主体的需求分析和面向问题框架<sup>[9]</sup>的建模方法等。这些研究应用逐渐广泛, 预计将在更多领域发挥重要作用<sup>[10]</sup>。OORA 是国内最早在需求工程中引入本体的建模研究, 虽然没有专门针对需求变更等管理工作做出指导, 但 OORA 模型提取需求对象基本元素: 角色、活动和数据, 使功能需求的分析建模容易理解、便于沟通, 使建模理论在需求分析活动中更容易应用, 成为有效的建模指导。ROB 需求模型在 OORA 基础上从三个不同视点抽取需求对象并整合为完整需求树。

(1) 业务视点。业务分解为一组更基本的业务及业务流; 业务流是业务二元有序对集合, 表示业务间流转关系; 更基本的业务基于递归思想可采用相同分解方法逐步分解形成树形结构; 最基本的业务称为活动, 表示为三元组: 活动(资源, 角色, Api)。Api 表示为五元组: Api(前置条件, 起始状态, 过程状态, 终止状态, 目标状态)。

(2) 组织视点。组织分解为一组规模更小的组织及辖制关系; 辖制关系是组织二元有序对集合, 表示组织间权限制约关系; 更小的组织可采用相同分解方法逐步分解; 最基本的组织即角色表示为一组权限的集合, 权限分为资源权限和业务权限, 资源权限是特定资源上一组操作权的集合, 业务权限是一组活动参与权的集合。

(3) 资源视点。资源分解为一组更基本的资源和资源关系; 资源关系是资源二元有序对集合, 表示资源间依赖关系; 基本资源采用相同的分解方法逐步分解, 形成资源需求树; 最基本的资源称为原子资源, 表示为结构和关系。

三个不同视点下分析结果: 业务树、组织树和资源树, 作为子树构成完整需求树(图 2)。

### 1.3 需求对象定义

ROB 模型中需求包括编号、版本、属性集合和业务、组织、资源; 业务包括一组属性、更基本的业务和流转, 更基本的业务是规模较小的业务, 规模最小的业务定义为活动; 活动包括一组属性、角色集合、资源集合和一个 API; 流转包括一组属性、前提活动和

结果活动; 组织包括一组属性、部门和部门关系; 部门包括一组角色及角色间关系; 资源包括一组属性、更基本的资源和约束, 更基本的资源是规模较小的资源,

最基本的资源成为元素; 元素包括一组属性和约束. 下面以巴科斯范式给出相关定义说明.

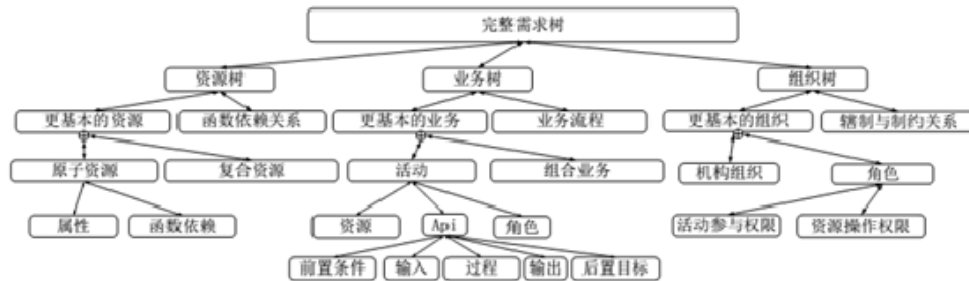


图 2 ROB 模型树形结构图

定义 1. 需求(Requirement)

Requirement::=(

Business::=( <Attribute>+,<SmallBusiness>+,<Circulation>+)

Circulation::=(

SmallBusiness::=(

Activity::=(

Organization::=(

Department::=(

Role::=(

Resource::=(

SmallResource::=(

Element::=(

ROB 需求对象具有统一的结构, 即树形结构, 与 Xml Schema 文法相对应, 采用 Xml Schema 的形式化描述如下.

```

<xs:complexType name="ReqBusiness">
  <xs:attribute name="BusinessID" type="xs:string"/>...
  <xs:element name="Activity" type="Activity"/>...
  <xs:element name="Circulation" type="CirculationArr"/>...
</xs:complexType>

```

```

<xs:complexType name="ReqOrganization">
  <xs:attribute name="OrganizationID" type="xs:string"/>...
  <xs:element name="Department" type="Department"/>...
  <xs:element name="DptRelation" type="DptRelation"/>...
</xs:complexType>
<xs:complexType name="ReqResource">
  <xs:element name="Attribute" type="AttributeArr"/>...
  <xs:element name="Constraint" type="ConstrainArr"/>...
</xs:complexType>
<xs:complexType name="Requirement">
  <xs:attribute name="Version" type="xs:string"/>...
  <xs:element name="Resource" type="ReqResource"/>
  <xs:element name="Organization" type="ReqOrganization"/>
  <xs:element name="Business" type="ReqBusiness"/>
</xs:complexType>...
<xs:element name="ReqObj" type="Requirement"/>

```

## 2 需求变更管理机制

### 2.1 需求变更管理指导

科学的管理需求变更是提高软件质量的重要保证.

CMMI 即能力成熟度集成模型,融合了全面质量管理(TQM)、ISO9000、六西格玛(6sigma)等作业体系,是结合项目管理方法和质量工程的产品研发管理方法,为软件过程改善提供有效指导<sup>[11]</sup>. CMMI 定义了 5 个组织成熟度级别<sup>[12]</sup>,包含 25 个过程域,涵盖了整个软件生命周期.需求管理属于成熟度 2 级过程域,包括 5 个特定实践,需求变更管理是过程域中一个特定实践,CMMI 对需求变更控制提出了明确指导:①建立需求变更管理组织;②制定需求变更控制流程;③使用软件工具管理需求变更.

### 2.2 需求变更管理组织与流程

需求变更管理是多种角色共同参与、各司其职、协同工作的过程,需求变更管理的实施依赖于管理组织与角色(表 1):变更控制小组(CCB)、产品部(PD)、项目部(PM)、软件部(SW)、测试部(TEST)、质量部(SQA)、软件质量保证(简称 SQA)、软件配置管理员(SCM)等.

表 1 需求变更管理组织与角色

部门	职责
CCB	1 审核变更需求(新需求,bug 修改,建议),2 确定处理方案
PD	1 定位产品.2 客户支持.3 提出需求变更申请到质量部
SQA	1 接收产品部申请.2 成立 CCB 小组,召集成员评审
PM	1 参与评审,确定可行性.2 变更通知到软件部和测试部
SW	1 可行性评估,编写报告(技术实现,进度,风险,建议).2 确定变更信息,制定计划,安排代码设计,更新需求说明书
TEST	1 参与评审.2 确定变更信息,制定测试计划,安排功能测试

多种角色协作对需求变更进行管控,依赖于 CMMI 指导下制定规范的需求变更管理流程(表 2).

表 2 需求变更控制流程

流程	部门	角色	任务	输出
申请	任	变更申请人	向 SQA 申请'需求变更'编号,填写提交给 SQA	需求变更申请单
组织	SQA	SQA	SQA 组织 CCB 小组,制定评审规则,会议议程	CCB 议程清单
分析	CCB	各部门主管	可行性分析,影响分析及工作量估算	分析报告
提交	CCB	各部门主管	CCB 签署'申请单',并同'评审纪要'和'分析报告'等到 PD; SQA 归档;	变更签署,分析评审纪要
变更	PD	PD 经理	接收 CCB 签署,分析评审纪要,确认需求变更发 PM	变更确认单
制定	PM	PM 经理	制定计划,评估可行性,将'变更通知单'申请单'发 SW	需求变更通知单
设计	SW	SW 经理,开发	编写可行性分析,实现方法.根据变更通知单,申请单安	变更实施方案

实施	SW	同上	人员	排设计	备份需求,协调一致的协同实施变更,版本发布	新版本需求
变更	TES	TEST			测试变更是否引起缺陷,遗漏等,验证是否完整,一致	变更验证
验证	T				记录变更状态,维护变更相关文档,度量分析	测试报告
归档	CCB	各部门主管				需求变更信息

### 3 需求变更管理数据模型

根据 CMMI 指导,实现需求变更自动化管理的直接手段是编制和使用配套软件(简称为变更软件).由于需求变更不可控,需求在软件全生命周期任何阶段都可能发生变更,不同需求变更之间相互影响而存在联系,变更软件目标是维护需求变更历史.根据定义 1,变更软件的需求包括:①业务,即需求变更管理流程;②组织,即需求变更管理组织;③资源,即需求变更历史记录.

实现需求变更自动化管理的核心问题是数据建模,即建立需求变更历史记录的数据存储模型,其目的是实现在计算机中存储与表示以便于对需求变更跟踪、回溯等.需求变更历史记录数据对象(ReqChgHistory Obj)是需求变更数据对象(ReqChgData Obj)的有序集合,每一个 ReqChgData Obj 表示一次成功实施需求变更的完整过程. ReqChgData Obj 包括:需求变更标识(ReqChgId)、需求变更属性集(ReqChgData Obj Attribute Set)、需求变更对象(ReqChgObj)及需求变更对象间的联系(ReqChgObj Rel).

定义 2. ReqChgHistoryObj

ReqChgHistoryObj ::= (<ReqChgData Obj>+)

ReqChgData Obj ::= (<ReqChgId>, <ReqChgData Obj Attribute Set>, <ReqChgObj>+, <ReqChgObj Rel>+)

ReqChgData Obj Attribute Set 表示对需求变更的描述,由一组关键字段组成,如申请组织、申请时间、申请原因、评估组织、评估时间、评估结果、审批组织、审批时间、审批结果等,这些关键字段根据变更管理组织、流程设置,此处为论述方便而给出简化的定义. ReqChgObj 集合和 ReqChgObj Rel 集合两个集合的组合表示需求变更的内容,其中每一个 ReqChgObj 表示变更中涉及到的某个特定需求对象,每一个 ReqChgObj Rel 则通过一对 ReqChgObj 之间的联系表示变更内容. ReqChgObj Rel 包括:1).变更联系(Sequence),由 ReqChgObj 二元有序对组成,表示从初

始需求变更对象(ReqChgObjFrom)变更为目标需求变更对象(ReqChgObjTo); 2.参照联系(Draw), 表示在特定情况(DrawObj)下, 需求变更对象(ReqChgObjFrom)参照需求变更对象(ReqChgObjTo)对待; 3.包含联系(Include), 表示实施变更后需求变更对象(ReqChgObjContain)包含另一需求变更对象(ReqChgObjIncluded); 4.交叉联系(Cross), 表示实施需求变更后, 需求变更对象(ReqChgObjCrS)与另一需求变更对象(CrossObj)有交叉. ReqChgDataObj 模型如图 3 所示.

ReqChgObjRel::=<ReqChgObjRelId>,<ReqChgObjRelName>,<ReqChgObjRelType>

ReqChgObjRelType::=<Sequence>|<Draw>|<Include>|<Cross>

Sequence::=<ReqChgObjFrom>,<ReqChgObjTo>

Draw::=<ReqChgObjFrom>,<DrawObj>,<ReqChgObjTo>

Include::=<ReqChgObjContain>,<ReqChgObjIncluded>

Cross::=<ReqChgObjCrS>,<CrossObj>

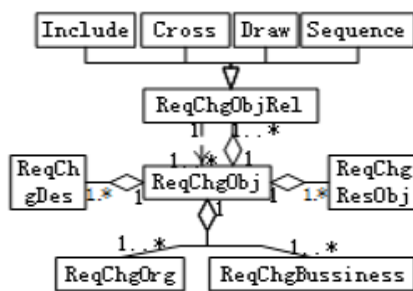


图 3 ReqChgDataObj 对象模型图

ReqChgObj 包含: 变更描述对象(ReqChgDes)集合、变更组织对象(ReqChgOrg)集合、变更资源对象(ReqChgRes)集合和变更业务对象(ReqChgBussiness)集合.

ReqChgObj::=<ReqChgObjId>,<ReqChgDes>+,<ReqChgOrg>+,<ReqChgRes>+,<ReqChgBussiness>+

ReqChgDes 用于描述需求变更对象, 分为结构描述(DesAttrib)和非结构描述(DesDoc). 非结构描述形式如文档、图片等, DesDoc 包含: 标识(DocId)、说明(DocStr)、格式(DocClass)和存储路径(DocPath). ReqChgDes 模型如图 4 所示, DesDoc、DesAttrib 与 ReqChgDes 间为特化关系.

ReqChgDes::=<DesAttrib>|<DesDoc>

DesDoc::=<DocId>,<DocStr>,<DocClass>,<DocPath>

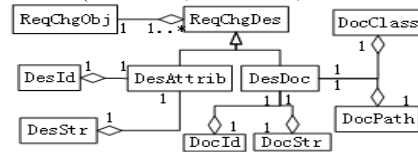


图 4 ReqChgDes 对象模型图

ath>

DesAttrb::=<DesId>,<DesStr>

ReqChgOrg 是一个特定组织对象, 用于描述发生变更的组织需求. ReqChgOrg 包括: 组织名(OrgName)、权限(Duty)、领导(Leader)、一般成员(Member)和更基本的组织(SmallOrg); ReqChgOrg 模型如图 5 所示.

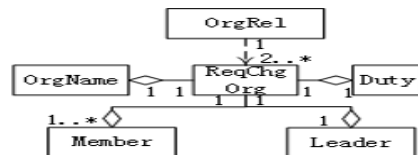


图 5 ReqChgOrg 对象模型图

ReqChgOrg::=<OrgName>,<Duty>,<Leader>,<Member>+,<SmallOrg>\*

Leader::=<Roles>

Member::=<Roles>

Roles::=<RoleName>,<Duty>

ReqChgRes 表示需求变更管理过程中使用或产生的各类资源, 包括: 角色资源(RolesRsc)、流程控制资源(CtrRsc)、需求数据对象资源(ReqObjRsc)、分析模型资源(ModelRsc)和文档报告资源(DocRsc). RolesRsc 表示变更管理过程中用于组建组织的资源; CtrRsc 是用于控制业务流程的资源; ReqObjRsc 包括需求和需求变更内容; ModelRsc 包括计算模型资源和度量元资源; DocRsc 表示需求变更管理过程中各种文档报告等. ReqChgRes 模型如图 6 所示.

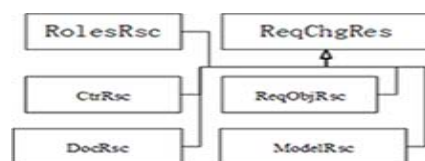


图 6 ReqChgRes 对象模型图

ReqChgRes::=<RolesRsc>|<CtrRsc>|<ReqObjRsc>|

<ModelRsc>|<DocRsc>

RolesRsc::=<(<RolRscId>,<RoleName>,<RoleRel>,<Duty>,<Identity>)

CtrRsc::=<(<CtrRscId>,<Period>,<Sequence>,<ProcDes>,<ProcState>,<Prcdure>)

ReqObjRsc::=<(<ReqRsc>,<ChgReqPrcRsc>)

ModelRsc::=<(<AnalysisMdlRsc>,<Analysis>)

DocRsc::=<(<DocRscId>,<DocType>,<DocDes>)

ReqChgBussiness 表示需求变更管理的整个过程,按时序分不同阶段进行,每一阶段执行不同任务,任务间有并行关系、时序关系和选择关系. ReqChgBussiness 包括: 变更业务名称(ReqChgBsnName)、业务描述(ReqChgBsnDes)、业务执行条件(SwitchCdt)和阶段任务集合(Period).

ReqChgBussiness::=<(<ReqChgBsnName>,<ReqChgBsnDes>,<SwitchCdt>,<Period>+)

SwitchCdt 包括业务开关规则(CdtChkFlw)和业务环境(Situation); Situation 由一组业务触发事件(Event)组成;

SwitchCdt::=<(<Situation>,<CdtChkFlw>)

Situation ::= <Event>+

Event::=<(<EventName>,<EventDes>,<EventAtt>)

Period 包括: 阶段名称(PeriodName)、阶段时序

(PeriodSequence)、阶段描述(PeriodDes)、阶段执行条件(SwitchCdt)和任务(Task)、任务时序(TaskSqc);

Period::=<(<PeriodName>,<PeriodSequence>,<PeriodDes>,<SwitchCdt>,<Task>+,<TaskSqc>+)

Task 包括: 任务名称(TaskName)、任务描述(TaskDes)、任务执行条件(SwitchCdt)和原子任务(AtomicTask)、原子任务时序(AtomicTaskSqc);

Task::=<(<TaskName>,<TaskDes>,<SwitchCdt>,<AtomicTask>+,<AtomicTaskSqc>+)

AtomicTask 有唯一标识,分六种类型(TaskType): 1).组织组建活动(CreaOrgAct),建立管理组织,分配任务职责; 2).流程制定活动(CreaPrcAct),提出计划,组织安排工作流程; 3).目标识别活动(DstgObjAct),在给定状态下,识别将进行和停止的目标活动; 4).业务控制活动(BussCtrAct),根据目标识别,控制执行相应活动; 5).度量分析活动(EvaluAnlAct),选用公式、参数给出度量结果; 6).基本任务活动(BasiTskAct),描述由哪些角色调度哪些资源输出哪些结果;

AtomicTask::=<(<AtomicTaskId>,<AtomicTaskName>,<TaskType>)

TaskType::=<(<CreaOrgAct>|<CreaPrcAct>|<DstgObjAct>|<BussCtrAct>|<EvaluAnlAct>|<BasiTskAct>)

ReqChgBussiness 对象模型如图 7 所示.

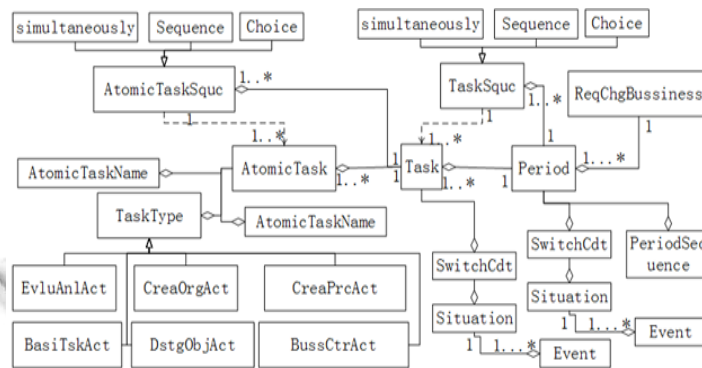


图 7 ReqChgOrg 对象模型图

### 4 需求变更管理数据操纵

#### 4.1 数据模型与接口定义

基于 ROB 的需求变更管理过程中所有对象统一为树型结构,采用 DOM 对象树模式存储需求变更对象,由于其结构灵活等特点,可表示具有丰富语义的需求变更对象; DOM 层次结构通过特化可转化为关系模式,采用关系模式操纵需求变更对象,可在关系规

范理论下定义、操纵需求变更对象.

为需求变更管理过程中所有对象定义操纵接口,规定基本 Api 操作,可以隔离底层数据存储模型以实现标准化操纵,例如为 reqChgObj 定义操纵接口:

```
interface reqChgObj extends Obj {
    string ReqChgObjId(); //系统指定
    string GetReqChgObjName(); //自定义属性...
```

```

boolean SetReqChgObjName(string);
boolean AddReqChgOrg(ReqChgOrg);
boolean DelReqChgOrg(ReqChgOrg);
boolean AddReqChgResObj(ReqChgResObj);
boolean DelReqChgResObj(ReqChgResObj);
boolean AddReqChgBuss(ReqChgBussiness);
boolean DelReqChgBuss(ReqChgBussiness);
boolean AddReqChgDes(ReqChgDes);
boolean DelReqChgDes(ReqChgDes);
Obj GetObj(Obj); //获取子对象接口
boolean SetObj(Obj); //设定子对象接口
}

```

由于具有统一的树形结构,可采用 Xml Schema 给出需求变更管理过程中所有对象的形式化定义,其目的是实现在计算机中的存储和管理.形式化方法与需求对象的形式化定义相同,本文不过多描述.

#### 4.2 需求变更操纵语言

由于需求变更管理对象全局结构统一,可定义统一操纵语言,规定相同的语法规则,实现对需求变更对象编辑修改,以完成需求变更管理任务.需求变更操纵语言基本组成时操纵语句,下面给出操纵语句(ManipulationStatement)定义.

定义 3. ManipulationStatement

```

ManipulationStatement::=<select>|<create>|<update>
|<delete>|<copy>
create::="create"(<class>,<ObjReference>,<Structure>)
delete::="delete"(<ObjReference>|"Cascade")
select::="select"(<ObjReference>)
update::="update"(<ObjReference>,<Structure>)
ObjReference::="<ObjId>|{"."<ObjId>"+
Structure::="("<SimType>|<ComType>)"
SimType::="(<ObjId>,<Type>)
Type::="Int"|"Dec"|"Char"|"Str"
ComType::="(<Member>+|<Connection>+)
Connection::="(<Member>,<Member>)
Member::="<Structure>
class::="Resource"|"Element"|"Attribute"|"Constraint"
|"Operation"|"OptConstraint"|"OperationRight"|"Role"|"
RoleRelation"|"Department"|"DptRelation"|"Organization"
|"Api"|"Activity"|"Circulation"|"Business"|"Int"|"Dec"|"

```

Char"|"Str"|<对象引用>

上述巴科斯范式描述需求控制操纵语言主要句法,包括:删除语句、更新语句、创建语句、查询/浏览语句.操纵语言字符集包括:52个字符(英文字符区分大小写)、数字、及/+-\*,>=<@!();等打印字符.

操纵语言语法规则主要有:1).ObjId(对象标识)自定义规则:由字符集中字符组成,不能与关键词重复;2).关键词:①命令保留字主要有“create”,“delete”,“Cascade”,“select”,“update”;②基本类型保留字包括“Int”,“Dec”,“Char”,“Str”;③内部类保留字包括“Requirement Change”,“Requirement”,“Resource”,“Element”,“Attribute”,“Constraint”,“Operation”,“OptConstraint”,“OperationRight”,“Role”,“RoleRelation”,“Department”,“DptRelation”,“Organization”,“Api”,“Activity”,“Circulation”,“Business”,“Int”,“Dec”,“Char”,“Str”.

操纵语言主要语句及成分包括:1).对象引用运算符“.”,通过ObjId表示对象间成员所属关系;2).需求对象引用表达式ObjReference,通过ObjId和对象引用运算符逐级引用需求对象;3).对象结构子句Structure,可以是简单结构子句或者复合结构子句;4).简单结构子句SimType,包括简单结构对象的标识ObjId和简单类型Type;5).复合结构子句ComType,包括一组成员Member表达式和一组关联Connection表达式的集合组成的复合表达式;6).成员表达式Member,由Structure子句构成;7).关联表达式Connection,由两个成员表达式组成,表示了一个二元关系;8).create语句:创建需求对象,<class>内部类标识了创建需求对象的类,ObjReference指出对象成员所属,Structure为对象结构说明;9).delete语句:删除需求对象,若带有Cascade命令参数,表示不删除ObjReference中ObjId标识的需求对象,同时将系统所有对象中的该对象成员删除;10).update语句:修改需求对象为一个新的结构Structure;11).select语句:定位、显示和浏览需求对象.

需求对象的数据操纵必须遵循完整性约束,包括:①域完整性,对象上可定义属性,满足对应规则,包括类型、格式、值域等;②实体完整性,任意需求对象客观存在且相互区别,具有唯一标识,且不为空;③参照完整性,任意需求对象的祖先或者子孙对象满足实体完整性;④根据实际的需求环境,用户通过制定特殊的规则可自定义约束.需求对象的数据操纵必须

满足域完整性、实体完整性和参照完整性约束。完整性约束是在需求变更控制模型上定义的一组规则,操纵语言必须严格遵守完整性约束规则才能保证数据的完整性、一致性。通过操纵语句可实现对需求对象的基本操作,进而完成需求变更控制过程中各个阶段任务,实现需求变更的科学管理。

## 5 结语

随着软件规模逐步扩大,需求分析活动不再仅限于软件开发的最初阶段,而是贯穿于软件整个生命周期。需求变更不可控,变更原因不统一,变更内容多样,对软件产生重要影响。

需求变更管理必须在科学方法和标准指导下,制定标准的管理流程,组建相应管理组织,协同一致的进行;需求变更自动化管理依赖于结构化、形式化程度,需求模型易于理解便于沟通则推动其应用。本文主要做出如下贡献:①基于递归、分解的思想提出 ROB 需求建模方法,从不同视角着手分析,容易理解便于沟通;②在 CMMI 指导下设计了需求变更管理流程和管理组织机构设置;③用巴科斯范式定义了需求变更管理过程所有对象,设计了需求变更管理模型;④定义了需求变更管理接口,并用 XML Schema 描述对象结构,设计了统一的数据操纵语言,从而实现了需求变更自动化管理。本文提取需求变更管理所有要素,抽取为规范化形式化对象,使需求变更管理在软件设计、开发与运行等各阶段有效。通过在实际应用中观察分析,这种基于 ROB 需求分析方法提出的需求变更管理模型在整个软件生命周期有效。

任何方法的提出需要实践中不断改进,下一步工作需通过一定数量的实验数据对比分析和改进、优化与完善需求变更度量体系、管理流程和管理组织;需求变更管理作为需求工程中一部分工作,必须与软件工程其他阶段工作协同一致,进一步工作需在软件工程全阶段改进或建立相应模型,使需求变更管理和软件开发各项工作相适应,以期形成完整的软件工程技术。

## 参考文献

- 1 Khoshgoftaar TM, Seliya N. Fault prediction modeling for software quality estimation: Comparing commonly used techniques. *Empirical Software Engineering*, 2003, 8(3): 255–283.
- 2 刘华琥,金英,马鹏飞.一种需求变更影响分析方法. *计算机研究与发展*,2013,50(8):1769–1777.
- 3 王健,刘衍珩,刘雪莲.复杂软件的级联故障建模. *计算机学报*,2011,34(6):1137–1147.
- 4 Damian D, Zowghi D, Vaidyanathasamy L. An industrial case study of immediate benefits of requirements engineering process improvement at the Australian center for unisys software. *Empirical Software Engineering*, 2004, 9(12): 45–75.
- 5 Katina PF, Keating CB, Jaradat RM. System requirements engineering in complex situations. *Requirements Engineering*, 2014, 19(1): 45–62.
- 6 于龙,蹇强.面向主题的信息抽取需求描述与分析. *计算机工程*,2012,38(23): 57–62.
- 7 Adar E, Teevan J, Dumais ST. The Web Changes Everything: Understanding the Dynamics of Web Content. *Proc. of the 2nd ACM International Conference on Web Search and Data Mining*. New York, USA: ACM Press. 2009. 282–291.
- 8 Kim N, Lee S, Moon S. Formalized entity extraction methodology for changeable business requirements. *Journal of information science and engineering*, 2008, 24(3): 649–671.
- 9 陈小红,尹斌,金芝.基于问题框架的需求建模:一种本体制导的方法. *软件学报*,2011,22(2):177–194.
- 10 Loughman TP, Fieck RA, Snipes R. A cross-disciplinary model for improved information systems analysis. *Industrial Management & Data Systems*, 2000, 100(8–9): 359–369.
- 11 卡耐基梅隆大学软件工程研究所编著.能力成熟度模型(CMMI):软件过程改进指南.刘孟仁译.北京:电子工业出版社,2001.
- 12 Agrawal M, Chari K, Fieck RA. Software effort, quality, and cycle time: A study of CMM level 5 projects. *IEEE Trans. on Software Engineering*, 2007, 33(3): 145–156.