

文章编号:1001-9081(2008)05-1287-04

一个支持软件需求验证的过程模型研究

荣辉桂^{1,2}, 李 珩¹, 郭卫锋¹

(1. 湖南大学 软件学院, 长沙 410082; 2. 武汉大学 信息管理学院, 武汉 430072)

(ronghg@163.com)

摘要:需求验证是软件需求阶段的一个重要环节,未经验证的需求给项目成功带来较大的需求风险。在前期研究的基础上,从需求验证的基本原理和可操作性出发,提出一个支持需求验证的过程模型(RVPM),进行形式化描述,并论述了需求验证过程的几个关键过程和策略。结合实例,分析了如何应用该模型来指导需求验证过程。理论和实例分析表明:该模型有效地克服了需求验证过程的复杂性和经验操作,有效降低项目需求风险。

关键词:需求验证; 活动序列; 任务分解; 对话图

中图分类号: TP311 **文献标志码:**A

Research of process model supporting for software requirements verification

RONG Hui-gui^{1,2}, LI Wei¹, GUO Wei-feng¹

(1. Software School, Hunan University, Changsha Hunan 410082, China;
2. School of Information Management, Wuhan University, Wuhan Hubei 430072, China)

Abstract: Requirements verification is an important procedure for software requirements, and requirements having not being verified will bring larger risk for the success of project. A Requirements Verification Process Model (RVPM) was proposed from the principles and operability of requirements verification, with reference to the former research of requirements analysis. Several key procedures and strategies of requirements verification were provided, and the corresponding formal description was given. Applying this model to guiding requirements verification was analyzed by a real example. Theoretical and experimental analyses show that this model can effectively overcome the complexity and experiential operation of requirements verification process and greatly reduce the risk of project requirements.

Key words: requirements verification; activities list; task division; dialog map

0 引言

经典的软件工程理论中,软件测试是代码完成后才开始,而事实上,测试应该从软件需求阶段开始。软件项目统计结果发现:50%以上的系统错误是由于错误的需求或缺少需求导致的,超过80%的开销花在追踪需求错误上,这是由于在追踪需求错误的过程中,容易混淆需求联系链,从而产生大量重复劳动^[1]。因此,项目在设计和编码前进行需求验证不仅必要,而且必不可少。

另一方面,需求的遗漏和错误具有很强的隐蔽性,仅仅阅读软件需求说明书,很难设想在特定环境下的系统行为。只有在业务需求基本明确,用户需求部分确定时,同步进行需求验证,才可能及早发现问题,从而在需求阶段以较低的代价解决这些问题。研究表明,需求分析阶段的错误如果不能得到及时纠正,将在后续阶段造成巨大代价,错误发现得越晚,修改的代价就越高。因此需求定义的准确性验证是一个重要课题。目前已经有一些自动化工具和系统化方法能部分支持需求定义内在的一致性和完备性检查^[2],使得软件需求的质量在一定程度上得到了保证,但需求定义是否准确地反映用户需求的验证在目前还主要依赖于文档评审和原型示范等技术,缺乏行之有效的系统化方法。本文首次提出支持需求验证过程模型,并定义支持需求验证过程的几个关键步骤,依据该模型指导,最终达到需求验证的目的。

文献[3]论述了需求层次模型,分析了基于用户需求进行用例建模,从用例中抽取功能需求条目等过程和策略。本文提出的需求验证过程模型基于该文献的一些前期研究工作进行,具体内容可以参考上述文献。

1 RVPM 过程模型

需求验证的核心是应用测试人员设计的需求测试用例来验证需求开发人员分析需求的成果,从而尽可能发现需求的遗漏、错误和二义性。基于该思想,本文将需求验证过程分解为功能需求条目获取、任务活动分解、基于用例的对话图建模、需求验证四个过程,首次提出软件需求验证过程模型(Requirements Verification Process Model, RVPM)概念;应用函数理论和过程控制方法确保 RVPM 过程的阶段结果满足完备性、一致性和正确性的要求,使得 RVPM 符合工程原则。

定义 1 RVPM 模型是以功能需求条目获取为基础,依据任务活动序列进行需求验证所要经历的所有阶段和步骤的集合,这些阶段可以用 $\langle Process_1, Process_2, \dots, Process_n \rangle$ 的过程集来表示,而每一个 $Process_i$ 是一个五元组 $\langle ID, SH_i, Res_i, Act_i, Cons_i \rangle$,其中: ID 是 $Process_i$ 的标识符; SH_i 是 $Process_i$ 所有可能的激励列表,一个激励序列是对系统的一系列单一输入; Res_i 是 $Process_i$ 的响应列表; Act_i 是 $Process_i$ 中所有活动的语义描述,定义了 $Process_i$ 中从激励到响应的所有映射规则; $Cons_i$ 定义了 $Process_i$ 的行为约束,通常包括执行 $Process_i$ 的初始条

收稿日期:2007-11-30;修回日期:2008-01-16。 基金项目:国家自然科学基金资助项目(60673093)。

作者简介:荣辉桂(1976-),男,湖南株洲人,讲师,博士研究生,主要研究方向:需求工程、服务工作流、电子商务; 李玮(1972-),男,湖南临澧人,讲师,博士研究生,主要研究方向:软件测试、嵌入式系统、可信网络; 郭卫锋(1976-),女,山西离石人,讲师,硕士,主要研究方向:软件工程、嵌入式系统、传感器网络。

件、前置条件和后置条件,可表达为 $\text{Cons}(\text{init}, \text{pre-con}, \text{post-con})$,而 init、pre-con、post-con 分别为初始条件、前置条件和后置条件集合。

通过定义可以给出 RVPM 模型描述过程的形式化描述,为了方便,一般用“元组元素”(ID)表示过程 ID 中的一类元素的集合,如 pre-con(A) 表示 Process_i 中 pre-con 元素的集合,那么获取需求 A 的功能模型分析过程可用如下形式化描述:

```
Type GetFunModel-A is ReqAnalyse
  Extern Receive ( SH : String)
  Public ResultS ( ReS : String)
  Action
     $\exists S_t \in \text{Stimulus}(\text{GetFunModel-A}) \wedge$ 
    Receive( GetFunModel-A, St)  $\Rightarrow$ 
    Results( GetFunModel-A, St)
  End GetFunModel-A
```

基于上述形式化定义和分析,结合工程实践,设计出 RVPM 模型,如图 1 所示。

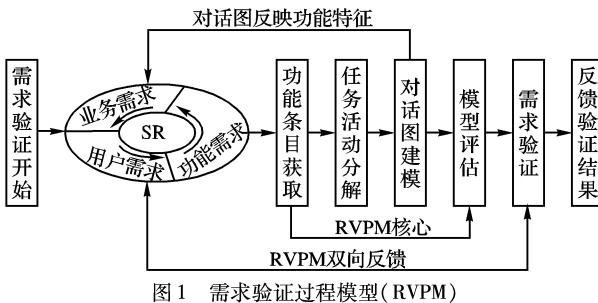


图 1 需求验证过程模型 (RVPM)

RVPM 需求验证过程模型中“SR”是软件需求 (Software Requirements) 的缩写,模型中椭圆示意部分说明业务需求、用户需求和功能需求三个层次需求间的联系,是 RVPM 模型的基础。模型通过“SR”部分和需求验证过程的双向反馈示意箭头表明需求验证过程是一个反复的过程,每轮需求验证定位了需求存在的问题后,应该将修订的需求内容尽快反馈到各级需求文档中,同时还应该进行确认验证,进一步确保需求验证的有效性。

2 RVPM 核心过程

RVPM 模型定义了从功能需求获取、任务活动分解、对话图建模到需求验证的四个核心过程:

- 1) 功能需求条目获取过程: 基于业务需求分析用户需求,对用户需求进行用例建模,从用例中获取功能需求条目;
- 2) 任务活动分解过程: 需求测试员先将选定的用例映射为一个独立的任务,然后将任务分解为粒度适中的活动集,子任务用集合中的有效活动序列进行描述;
- 3) 对话图建模过程: 需求分析员基于选定用例和功能需求条目设计对话元,进行对话图建模;
- 4) 需求验证过程: 需求测试员将分析得到的若干任务活动序列“概念执行”需求分析员分析得到的对话图模型中所有可能对话路径,期望找到两者不一致的描述,这些不一致也恰恰反映了基于选定用例的需求定义可能产生歧义、遗漏、多余等不一致。

2.1 功能需求条目获取

需求工程领域通常认为,功能需求是用户利用相关需求约定来完成任务、满足业务需求的要求,它描述了开发人员需要实现的具体内容,也是必须在产品中实现的软件功能。产品特性或者特征,则是指一组逻辑上相关的功能需求,它们为用户提供某项功能,使业务目标得以满足。

功能需求获取过程是 RVPM 模型中定义的第一个核心

过程,它是设计任务活动序列和对话图模型的基础。文献 [3] 提出了一个较好的功能需求获取过程,并结合实例分析具体的可操作过程,本文以此作为 RVPM 需求验证过程模型的重要组成部分,在此基础上较好地进行后续的任务活动序列设计和对话图建模,限于篇幅,在此仅作简单介绍,具体内容可以参考该文献。应用该文献中所描述的需求获取过程,方便获取良定义的功能需求,为需求验证过程作好准备。

2.2 任务活动分解和形式化描述

一个用例模型中包含多个用例,一个单独的用例实际就是一个相对独立的任务,粒度划分合适的用例恰恰可以分解为数量合适的子任务^[4,5]。

单个任务从入口到出口可以描述成一个完整任务的环节,形成一条从入口到出口的路径,它由一序列处理事件和反映特殊情形的谓词组成,从而完成一项特定的计算机功能。

一个任务描述由一序列活动组成,每项活动可以用如下的结构来表示:

基于该活动描述结构,用 A(Activation) 来描述活动,用 E(Execute) 描述动作执行,用 O(Object) 描述对象,用 D(Do) 表示动作,约束条件则用 C(Constrain) 进行描述。则上述结构可以描述如下:

$$\langle A \rangle := \langle E \rangle \langle D \rangle \langle O \rangle \langle C \rangle$$

假设一个任务以 Task 表示,那么一个任务包含的活动集描述为:

$$Task = \{A_1, A_2, \dots, A_k\}$$

一个任务中包含一个活动集合,而活动集合中,依据一定业务规则,相关联的活动的有序排列就形成一项子任务,实现特定的目标。子任务中的活动有序排列组合可能有多个,描述为 $SUBTask_{(i)} = \{A_m \rightarrow A_n \cdots A_q\}$ 。其中 i 为任务下辖子任务编号,m, n, q 为该任务活动集中所属活动的编号。

因此,一项任务如果包含 n 个活动,则其最多可以分解为式(1):

$$\text{SUM}(SUBTask)_{\max} = \sum_{k=1}^n p_n^i; i = 1, 2, \dots, n \quad (1)$$

其中 n 是指 Task 任务中所有活动的数目,SUM(SUBTask)_{max} 为所有活动序列的数目,也即是指最大子任务个数。

SUM(SUBTask)_{max} 表示一个任务中理论上最大可能的活动路径数目,然而并非所有的活动路径均为有效活动路径,有效路径必须是满足特定的功能需求,并得到客户认可的路径,其定义如下:

定义 2 有效活动路径是指包含在用例描述中,且满足相关业务规则的活动执行路径,每条执行路径的完整描述即可理解为一个子任务的执行。

因此,任务分解只需要包含有效活动路径。

尽管活动的逻辑定义清晰,然而实际用户需求分析中,基于用例将任务分解为活动并非凭经验来划分,而是依据一定的划分策略,活动划分必须满足以下要求:

- 1) 每一个活动参照上述定义的活动结构来分解,必须符合 $\langle A \rangle := \langle E \rangle \langle D \rangle \langle O \rangle \langle C \rangle$ 。
- 2) 活动分解应该参照对话图模型中对话图元,分解时必须维持同一分解粒度。
- 3) 活动序列必须有意义,且为一个有向序列,该有向序列能完全恢复子任务功能。
- 4) 针对特定任务分解得到的活动集合应该满足充分性和完备性。

任务分解完毕后,需要将任务中活动集所包含的活动,与

对话图模型中对话图元进行映射。在设计活动结构时,其包含的三要素恰恰对应对话图模型中导航路径和所连接的两个对话图元,在一定的映射规则下,保证活动和对话图元语义等价。

2.3 对话图建模语义分析

对话图将用户界面视作一个有限状态机,在某一个时刻只有一个对话元素可以接受用户输入^[6]。在激活的输入区中,用户可以根据自己所采取的动作导航到某些其他的对话元素。在一个相对较复杂的图形用户界面中,可能的导航路径数目会很大,其选项通常也是可知的。对话图的状态转换反映了功能需求描述的动作执行路径,为进行需求验证提供了依据。

基于 UML 规范的对话状态项描述:

UML 的状态包括简单状态、非并发复合状态和并发复合状态、初始状态、最终状态和子状态机状态、历史状态、同步状态等。

假设 UML 对话图状态机的状态名空间为 N , 其中 $initial_s \in N$, $final_s \in N$, 分别表示状态项 S 的初始状态和最终状态。整个状态机中变迁的集合用 T 表示, UML 对话状态项的代数定义如下:

定义 3 UML 对话状态机的状态项集 SA 是由以下规则所定义的最小集合:

1) 空间态 \varnothing : 是一个状态项, 它只是表示对话状态机的初始状态的父状态。

2) 基本状态: 对任意 $n \in N$, $s_p \in SA$ 是状态 n 的父状态项, 则 $s = \lfloor n; s_p; EntryAction; DoActivity; ExitAction \rfloor$ 是一个状态项, n 即是指该状态的名字, $EntryAction$, $DoActivity$, $ExitAction$ 分别表示该状态的入口动作、活动和出口动作, 由函数 $Entry: SA \rightarrow Action$, $Do: SA \rightarrow Action$ 和 $Exit: SA \rightarrow Action$ 可分别得到三个动作, 对 $n \in SA$, $Entry(s) = EntryAction$, $Do(s) = DoActivity$, $Exit(s) = ExitAction$, 函数 s 表示状态 n 的状态项。

3) 与状态: 如果 $n \in N$, $s_p \in SA$, $k > 0$, SA 中的项 s_1, s_2, \dots, s_k 是状态 n 正交区域, 是状态 n 的并发子状态, 而 s_p 是状态 n 的父状态项, 则有 $s = \lfloor n; s_p; (s_1, s_2, \dots, s_k) EntryAction; DoActivity; ExitAction \rfloor$ 是状态 n 的一个状态项。

4) 或状态: 如果 $n \in N$, $s_p \in SA$, 而 s_p 是状态 n 父状态项, $s = \lfloor n; s_p; (s_1, s_2, \dots, s_k); l; T_n; h; D_h; EntryAction; DoActivity; ExitAction \rfloor$ 是状态 n 的状态项, $k > 0$, SA 中的项 s_1, s_2, \dots, s_k 是状态 n 的子状态; $l \in \{1, 2, \dots, k\}$, 表示 s_l 是状态 n 的当前活跃子状态; T 是状态 n 中连接各子状态的所有变迁的集合; $h \in \{\text{none}, \text{shallow}, \text{deep}\}$ 是一个标记, 表示该状态或状态是否拥有历史状态, none 表示没有, shallow 表示有个浅历史状态, deep 表示有个深历史状态; $D_h \in \{1, 2, \dots, k\}$, 表示 S_{D_h} 默认的历史状态。

对话状态的精确形式化描述较好地表达了 UML 对话状态的动态语义, 也是对话图建模的理论基础。限于篇幅, 状态变迁的描述和证明文中不再赘述。

对话图模型是用户需求建模的重要成果之一, 它以图形化的方式, 清晰地表达了用户需求内容, 因此, 基于用例的对话图建模被纳入 RVPM 模型的核心。

2.4 需求验证过程

基于 RVPM 模型中的定义的需求验证原理, 设计需求验证具体过程, 如图 2 所示。

需求验证过程中, 用例和功能需求的获取是基础。一轮完整的需求验证过程描述如下: 获取用例和功能需求条目; 对

选定的用例进行对话图建模, 寻找界面导航的可能路径; 需求测试人员根据任务活动序列所描述的若干子任务在对话图中所有可能的导航路径上进行“概念上”的执行, 以期发现遗漏的、错误的和不必要的需求; 根据需求验证结果修改各级需求文档, 直至测试用例覆盖所有的用例和功能需求条目为止。

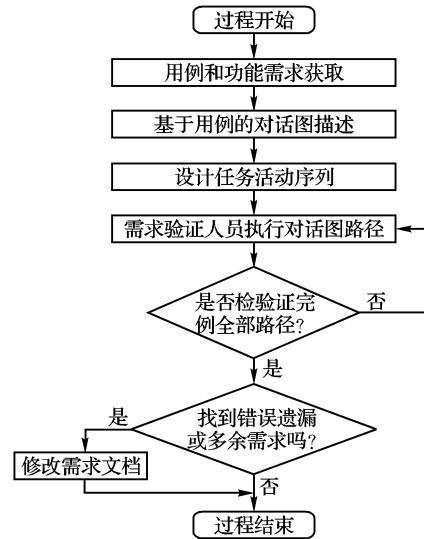


图 2 需求验证过程

3 应用 RVPM 模型指导需求验证

本节重点论述应用 PVPM 模型指导实际项目进行需求验证的过程。围绕“网络书社”项目中与“请求购买某本书籍”这一任务相关的一个用例进行分析。因网上购书的流程比较单一, 许多读者都有过类似网上购书的经历, 该项目完整的用户需求和用例分析就不在文中赘述。

3.1 基于用例生成对话图模型

依据文献[3]中定义的需求获取过程, 从网络书社项目中分析得出“请求某本书籍”的业务需求、用户需求和功能需求描述如下:

1) 业务需求 BR-1: “网络书社”最主要的业务目标之一如下描述: 传统的柜台销售收入占公司下辖连锁书店和加盟小书店利润的较大比例, 项目完成后, 这一比例可能发生变化, 网上定书业务量会大量增加, 同时客户种类也将在传统的固定批量订购客户基础上增加个人网络客户。

2) 用户需求 UR-1: 支持上述业务目标的一个用例 UC-1 是“请求指定的书籍”, 该用例描述如下: 允许用户请求书社中(含个体小书店)库存中有货的书籍。下面是该用例的详细说明:

请求者输入书籍名称或作者, 系统则从书社数据库中拨给请求者一本新书, 或者让他提交一个从外部出版社定购的请求, 从而满足其购书要求。

3) 功能性需求 FR-1: 以下是与用例 UC-1 对应的功能说明: 如果书社数据库中有所请求的书籍, 系统显示库存中有该书的所有书店; 用户从所有列表显示的书店中选择其中一个书店下订单, 或者向外部出版社定购一本新书。

以对话状态的描述语义为基础, 根据对话图建模过程, 得到基于该用例 UC-1 的对话图 DM-1, 如图 3 所示。

图中的矩形框表示用户界面画面, 箭头则表示从一个画面切换到另一个画面的可能导航路径。对话图 DM-1 很好地描述用例中外部执行者通过对话图元和系统进行交互的过程。

程,它从较高抽象层次上表示用户界面的设计,并展示了系统的对话元素及这些元素之间的导航连接,连接之间的切换是通过执行用例中描述的功能动作完成。因此,基于用例的对话图建模通常用来作为需求图形化和需求建模的成果。

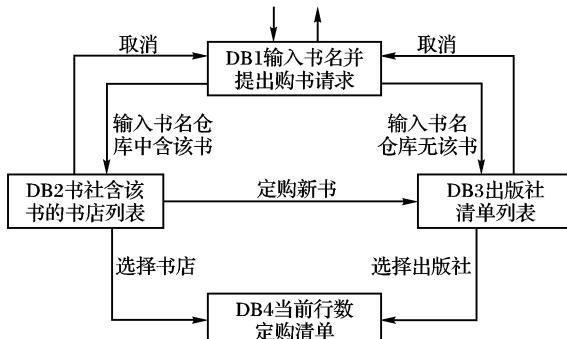


图 3 “请求某本书”用例对话图 DM-1

根据已建模的对话图模型 DM-1,分析得到四个对话图元:

$DB1 = \{输入书名, 并提出请求\}$; $DB2 = \{书社有该书的书店\}$; $DB3 = \{出版社清单\}$; $DB4 = \{当前新书定购清单\}$ 。

如果两对话图元之间存在有向的对话连接,即表明在一定条件下,可以从前述对话图元状态切换到该对话图元状态,该对话连接称为对话图的导航路径。四个对话图元可能产生导航路径 P_n^2 条,也即有 P_4^2 共 12 条,但是实际上有效导航路径为 5 条,接近或小于 C_4^2 共 6 条。

3.2 基于活动序列的子任务设计

需求验证设计的任务分解策略,活动序列路径应该能覆盖用例的主干过程、分支过程和在需求获取和分析期间所确定的异常条件^[7]。“网络书社”项目中,对用例 UC-1“请求指定书籍”进行分析,得到基于用例 UC-1 的任务所含的全部活动集为 {A1, A2, A3, A4},活动元素对应的内容描述如下:

A1=(输入合法书籍名称); A2=(显示含请求书的书店); A3=(加入出版社清单列表); A4=(写入定购清单列表)。

考虑该用例的若干可能的执行路径,容易设计出若干子任务来分别处理主干过程、分支过程和异常。基于用例 UC-1 和对话图 DM-1 设计出以下子任务描述过程:

$SUBTask_{(1)}$: 在 DB-1 对话框中,输入一个合法的书籍名称;书社中有两个书店包含该书籍。此时弹出 DB-2 对话框,并显示这两个书店。选择第二个书店购买该书,关闭 DB2,二号书店中购书信息添加到 DB-4 对话框中显示的购书订单底部。

$SUBTask_{(2)}$: 在 DB-1 对话框中,输入一个合法的书籍名称;书社找不到该书籍。此时弹出 DB-3 对话框,并显示含该书的出版社信息。选择一个出版社请求该书,关闭 DB3,从出版社定购该书信息被添加到 DB-4 对话框中显示的购书订单底部。

$SUBTask_{(3)}$: 在 DB-1 对话框中,输入一个合法的书籍名称;书社中有两个书店包含该书籍,弹出 DB-2 对话框。此时,若临时取消该书籍的定购,则回到 DB-1 对话框界面。

基于该用例还有其他子任务,如分支过程的取消等可能路径,本文中不详细列出。

结合 UC-1 所涉及活动集的 4 个活动,根据业务需求规则,上述三个子任务的活动序列可描述如下:

$$SUBTask_{(1)} = \{A1 \rightarrow A2 \rightarrow A4\}$$

$$SUBTask_{(2)} = \{A1 \rightarrow A3 \rightarrow A4\}$$

$$SUBTask_{(3)} = \{A1 \rightarrow A2 \rightarrow A1\}$$

值得注意的是:上述用于需求验证的子任务是需求测试人员依据用例和功能需求设计的,尽管验证的活动序列和对

话图模型的设计对象不一致,但它们都来源于同一个基础——用户需求^[8]。

3.3 简化的需求验证过程

根据需求验证过程模型 RVPM,应用上述子任务活动序列描述,来进行需求验证。

$SUBTask_{(1)}$ 验证过程:根据验证子任务活动序列 $SUBTask_{(1)}$ 描述内容,用虚线条在对话图 DM-1 中跟踪子任务的执行路径。在 DB-1 对话框中,输入一个合法的书籍名称;书社中有两个书店包含该书籍。此时弹出 DB-2 对话框,并显示这两个书店。选择第二个书店购该书,关闭 DB2,二号书店中购书信息添加到 DB-4 对话框中显示的购书订单底部。图 4 中的虚线箭头描绘了在“请求一本书籍”用例对话图中跟踪子任务活动序列 $SUBTask_{(1)}$ 的执行路径。

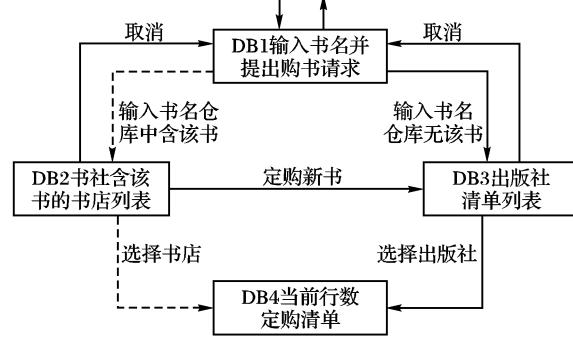


图 4 子任务活动序列“执行”示意图

同样, $SUBTask_{(2)}$ 、 $SUBTask_{(3)}$ 的验证过程也类同,通过跟踪每个子任务活动序列的执行路径,验证人员可能发现和需求分析人员对需求不一致的理解,从而找出不正确的、遗漏的或不必要的需求,在对话图和子任务中加以纠正,并修改相应的需求文档。

如果以上述方式“执行”完所有子任务活动序列后,对话图中 DB2 到 DB3 之间标有“定购新书”的导航线仍未被加亮,那么存在如下两种解释:

1) 从 DB2 到 DB3 的导航是一个非法的系统行为。分析人员应该将这条线从对话图中删除,并且如果软件需求规格说明书中也包含这条指定状态转换的需求,那么也必须从中将这条需求删去。

2) 该导航是合法的系统行为,但是遗漏了展示这一系统行为的子任务描述,如果软件需求规格说明书中未包含这条指定状态转换的需求,那么必须将这条需求加入。

类似,假设用户可以采用某些动作直接从 DB1 对话框转换到 DB4 对话框,而图 4 中对话图 DM-1 中并没有包含这样的导航线,所以,子任务活动序列不能以现有的需求来执行。上述情况存在两种解释,需要判断,本文不再展开论述。

然后,逐步纠正各级文档和模型中需求描述和定义存在的问题,完成一轮需求验证过程。因此,以 RVPM 模型为指导,可以很好地进行需求验证过程,最终实现全部需求验证。

4 结语

本文提出一个支持需求验证过程模型 RVPM,分析了该模型的结构,论述了该模型定义的四个核心支撑过程,并结合项目实例,应用该模型进行具体需求验证过程分析。分析表明:应用 RVPM 模型可以指导需求测试人员快速进行需求验证,便于在项目早期发现需求的遗漏、错误和冗余,并加以纠正;该模型有效地克服了需求验证过程的复杂性和经验操作,大大降低项目需求风险。
(下转第 1294 页)

后将两者绑定发送并备份。

2) 验证阶段:接收端 ambox 收到邮件后提取其中的邮件部分,计算其“消息摘要”,然后与其携带的“消息摘要”进行比对,如果相同则置 IsBad 和 IsDispose 为 0;若不同,则丢弃此邮件,并置 IsBad 为 1。

3) 处理阶段:MI2 的 ambox 注册时 pmbox 将列表记录的 IsBad 为 0 邮件重新放入发送缓冲池等待发送,并置邮件的 CheckNum 和 IsSend 为“0”。

以上是两类异常通信情况的处理,当邮件重新放入缓冲池,由于其序号较小,会提前被发送出去,使 MI 能够尽早处理。

4 邮件保序性

邮件保序是指邮件可以按照创建时间或者依赖关系次序得到发送和处理。定义 4 中 mail 的 DependOn 字段,用于邮件的保序。默认值为 null。当 DependOn 的值为 null 时,采用“发生在先”^[7]机制,这一机制体现在 pmbox 和 ambox 的“邮件序号最小者优先操作”;当 DependOn 的值为一个或几个直接前驱邮件名时,即为“因果保序”^[7]。下面对邮件转发算法进行补充以支持保序性。

解决方案:在该邮件转发算法中,有两处要处理邮件的保序性,一是 pmbox 端邮件发送,二是目标 MI 从 ambox 提取邮件处理。针对这两种情况的方案有所不同,表现在邮件的前驱依赖,前者只要等待发送的邮件队列中没有此邮件的前驱邮件就可以把此邮件发送出去,后者则必须在此邮件的前驱邮件全部得到处理才能把此邮件提交处理。

1) pmbox 端邮件发送保序算法

```
For(选出序号最小的邮件,记为 M)
{ numchange = true;           //检验是否有序号交换操作
  if( M-DependOn = null) { send M; continue; }      //发送邮件 M
  While( M-DependOn-link != null)
    //还未检索完 DependOn 记录 M 的前驱邮件
    { 提取当前指针指向的邮件标识,与 pmbox 的邮件列表中
      IsSend = 0 的邮件进行比对;
        if( 找到一个邮件记为 N 与此邮件标识相同)
          { 交换 M, N 的邮件序号; numchange = false; break; }
          M-DependOn-link -> next;                  //转到下一条记录
      }
      if( numchange = false) { continue;
    }
    send M;
}

2) MI 从 ambox 提取邮件进行处理
For(选出序号最小的邮件,记为 M)
{ numchange = true;           //检验是否有序号交换操作
  already = true;             //检验前驱邮件是否已经都被处理
```

(上接第 1290 页)

参考文献:

- ```
if(M-DependOn = null) { dispose M; continue; }
While(M-DependOn-link != null)
 //还未检索完 DependOn 记录 M 的前驱邮件
 { 提取指针指向的邮件标识,与 ambox 的邮件列表中 IsBad = 0
 的邮件进行比对;
 if(找到一个邮件记为 N 与此邮件标识相同)
 { if(N-IsDispose = 0)
 { 交换 M, N 的邮件序号; numchange = false; break; }
 }
 else{ already = false; break}
 M-DependOn-link -> next; //转到下一条记录
 }
 if(numchange = false) { continue; }
 if(already = true) { dispose M; }
 else{ 搁置 M 一定时间间隔 To 然后重新放入缓冲区}
 }

5 结语
本文首先在基于迁移计算范型的迁移工作流系统上,提出一种全新的 MI 间消息通信机制,并给出详细的邮件转发处理算法。按照此种通信机制转发邮件不需要寻址,邮件即可直接发往注册地址,命中率相对要高,而且在 MI 迁移过程中邮件的转发并未停滞,可以使邮件尽快到达 MI。其次,本文还分析了邮件转发过程所出现的各种异常情况(其中邮件丢失、数据丢失、消息篡改等异常以往通信算法未能给出有效的解决方案),给出了相应的解决方案并在算法中得到实现。最后,本文分析了邮件的保序性,并在 pmbox 和 ambox 中采用不同的算法策略实现。
```
- 参考文献:**
- [1] 曾广周,党妍. 基于移动计算范型的迁移工作流研究[J]. 计算机学报, 2003, 26(10): 1343–1349.
  - [2] 陶先平. 基于 Internet 的移动 Agent 技术和应用研究[D]. 南京:南京大学, 2001.
  - [3] FENG XIN-YU, CAO JIAN-NONG, LÜ JIAN, et al. An efficient mailbox-based algorithm for message delivery in mobile agent system [C]// Proceedings of the 5th International Conference on Mobile Agents, LNCS 2240. London: Springer-Verlag, 2001: 135–151.
  - [4] 周竟扬,陈韬略,陈道蓄,等. 一种高效可靠的移动 Agent 间通信机制[J]. 软件学报, 2003, 14(08): 1470–1480.
  - [5] 陶先平,冯新宇,李新,等. Mogent 系统的通信机制[J]. 软件学报, 2000, 11(8): 1060–1065.
  - [6] 王红,曾广周,林守勋. 可移动 agent 系统位置透明通信的一种实现[J]. 计算机学报, 2001, 24(4): 442–446.
  - [7] 陈韬略,韩亮,陶先平,等. 移动 Agent 系统的一个因果保序通信协议[J]. 计算机工程, 2005, 31(7): 120–121.
  - [8] WEIGERS K E. 软件需求[M]. 刘伟琴,刘洪涛,译. 2nd ed. 北京:清华大学出版社, 2006.

成[J]. 计算机学报, 1998, 21(8): 673–681.

- [6] LIN Y C, CHIANG T H, WANG H M, et al. The design of a multi-domain mandarin chinese spoken dialogue system[C]// Proceedings of 5th International Conference on Spoken Language Processing (ICSLP'98). Sydney: [s. n.], 1998: 41–44.
- [7] DURÁN A, RUIZ-CORTÉS A, CORCHUELO R, et al. Supporting requirements verification using XSLT[C]// Proceedings of the 10th Anniversary IEEE Joint International Conference on Requirements Engineering. Washington, DC: IEEE Computer Society, 2002: 165–172.
- [8] WEIGERS K E. 软件需求[M]. 刘伟琴,刘洪涛,译. 2nd ed. 北京:清华大学出版社, 2006.