

# 面向行为的需求建模研究及实现

万黎 毋国庆 吴怀广

(武汉大学计算机学院 武汉 430072)

**摘要** 需求建模是需求工程中的重要步骤。提出了一种面向行为的需求建模方法,首先将目标系统问题域划分成多个子问题域,然后在子问题域里建立多个视点,在每个视点内,根据需求建立多个场景。对于每个场景,用行为描述语言建立场景行为模型,视点内所有场景的行为模型组合构成视点行为模型,目标系统所有视点的行为模型组合构成系统行为模型。在行为模型的基础上,提出了一系列特性以及其验证方法,用于验证行为模型的正确性以及系统特性。基于以上方法,设计和实现了面向行为的需求建模工具(BRMT),以辅助开发人员从行为的角度进行需求建模。

**关键词** 需求建模,软件行为,视点,场景,行为描述语言

中图分类号 TP301 文献标识码 A

## Research and Implement of Behavior-oriented Requirements Modeling

WAN Li WU Guo-qing WU Huai-guang

(Computer School, Wuhan University, Wuhan 430072, China)

**Abstract** Requirements modeling is a key procedure in Requirements Engineering. This paper put forward a behavior-oriented modeling methods: Plotted out problem domain of target system into several sub-domains. Viewpoints were then created in each sub-domain. Within each Viewpoint, a series of scenarios would be established according to requirements. A Scenario Behavior Model(SCBM) could then be established to specify each scenario by Behavior Description Language(BDL). A Viewpoint Behavior Model(VBM) could be constructed of all SBM within the Viewpoint and a System Behavior Model(SBM) could be constructed of all VBMs. A set of properties and their verification methods were proposed to check the correctness of behavior model and properties of the target system. Based on the methods above, a Behavior-oriented Requirements Modeling Tool(BRMT) was implemented.

**Keywords** Requirements modeling, Software behavior, Viewpoint, Scenario, Behavior description language

## 1 前言

在软件开发的实践过程中,人们越来越清楚地认识到软件需求工程是软件开发的瓶颈。Standish 集团公司在 2009 年的软件项目调查报告显示,有 32% 的软件项目能够在规定的时间和预算内成功完成;44% 的项目正遭受着延期、预算超支、以及/或者不能完全满足预期特性和功能的问题;有 24% 的软件项目已经宣告失败<sup>[1]</sup>。Standish 集团公司的另一项研究分析显示,项目失败或严重超支的 8 个重要原因中有 5 个都与需求相关:即需求不完整,缺乏用户参与,客户期望不实际,需求和需求规格说明的变更和提供许多不必要的功能<sup>[2]</sup>。因此,如何获取高质量的软件需求成为需求工程的主要目的。需求工程由一系列与软件需求相关的活动组成,其中一个重要过程是需求建模。需求建模的工作就是导出目标系统的需求模型。借助于需求模型,开发人员可以更精确地表达和理解需求,检验系统的一些特性,并且可以更好地与用户沟通,确认需求是否满足用户的期望。

无论使用什么样的需求建模方法和技术,以及其它的软

件开发方法和技术,最终目的都是希望能开发出高质量和满足用户需求的软件。而软件是否能满足用户需求,这又是通过实际执行的软件行为所决定的。因此,软件行为的正确与否决定了软件能否满足用户需求。此外,软件特性如可信特性也是通过软件行为来验证的,但在需求阶段由于软件还未开发出来,故要验证软件特性是相当困难的工作。如果能在需求阶段通过分析待开发软件的行为,并建立严格的描述基于软件行为的需求模型,将是非常现实和重要的工作。

很多建模方法都涉及到了软件的行为。在形式化的方法中,进程代数用代数算子表示系统的行为和动作序列,比较典型的有用于描述通信并发系统行为的通信系统演算 CCS<sup>[3]</sup>和用于描述分布式系统行为的通信顺序进程 CSP<sup>[4]</sup>。时序逻辑通过逻辑算子和时序算子,表达系统中动作的逻辑和时序关系<sup>[5]</sup>。Petri 网中使用原子变迁来表示软件的行为<sup>[6]</sup>。这些建模方法可以建立准确、无二义性的模型,并且容易验证模型的正确性。但建立的模型大都使用严密的数学符号来表示,开发人员往往需要很长的时间才能掌握,软件需求的用户也难以理解这些模型,这就是形式化应用推广的瓶颈。

收稿日期:2010-05-26 返修日期:2010-08-19 本文受国家高技术研究发展计划(863 计划)(2007AA01Z185)资助。

万黎(1983-),男,博士生,主要研究方向为形式化方法、软件需求工程,E-mail:wllwxz@gmail.com;毋国庆(1954-),教授,博士生导师,主要研究方向为形式化方法和软件需求工程;吴怀广(1976-),男,博士生,主要研究方向为进程代数理论、软件需求工程。

在半形式化方法中,以 UML 为代表的面向对象的分析方法<sup>[7]</sup>将问题域中的事物建模成为对象,使用分类、泛化、聚类等方法组织这些对象,然后研究这些对象间的交互行为,如 UML 中的时序图、状态图。UML 的这种半形式化图形表示法易于被开发人员掌握,并且模型的可读性强,然而 UML 缺乏严格的语义,很难验证模型的正确性。

国内在软件需求建模方面经过多年的研究和实践,除了对上述方面的推进研究外,还取得了不少成果和进步。文献[8]提出了面向本体的需求分析方法,提出以本体作为对象关联手段,从而加强面向对象方法的表达能力。另外,在本体概念的基础上,文献[9]介绍了面向服务的建模方法。文献[10]给出了一种特征模型的具体形式,并结合具体的领域,对其建模过程进行了详细论述。

基于上述分析,我们提出了一种直接描述软件行为的形式化语言——行为描述语言(BDL)<sup>[11]</sup>。在 BDL 中,待开发软件系统中的一个简单操作,状态变化可以被描述为一个细粒度的原子行为,原子行为之间有数据传输。原子行为可通过逻辑符号组合成粗粒度的复合行为。复合行为也可以通过逻辑符号组合成更粗粒度的复合行为。BDL 语法简单,可读性很强,开发人员可以很快掌握并用其建模,建立的模型也很容易被普通的用户读懂,有助于开发人员和用户进行沟通。

BDL 能够很好地描述软件系统的执行过程。然而,在一个复杂的软件系统中,庞大的软件规模和复杂的内部逻辑关系使建模问题异常复杂。另一方面,由于复杂系统往往涉及到许多相关人员,这些人员由于各自的背景知识、职责及出发点不同,因此会从不同的角度和立场来提出自己的观点与需求,从而形成不同的需求。为获得复杂系统的高质量和可信需求模型,需根据不同的用户视点建立系统的需求模型。为此,我们基于 BDL 设计了一套建模方法,首先对目标系统进行自顶向下的分解,然后用 BDL 自底向上地为目标系统建立行为模型。我们将待开发软件系统的问题域划分成多个问题子域,并且借鉴多视点的思想,在这些问题子域内创建一个或多个视点<sup>[12,13]</sup>,每一个视点用一个视点行为模型来表示。在每个视点内,根据待开发软件的需求,划分多个场景,以场景为单位组织单个执行序列,每一个场景用一个场景行为模型来表示。这样,可为系统建立完整的行为模型。

在需求模型的构建过程中,各视点虽然是相对独立的,但不同视点间的需求信息可能会发生重叠。因此,由不同视点产生的需求模型间会产生重叠和差异,从而导致需求模型间可能会出现需求冲突和不一致。此外,有些视点可能使用不同的需求建模方法和技术构建需求模型,更增加了视点间容易发生需求冲突和不一致的可能性。因此,在形成最终需求规约之前,必须检测和处理视点间存在的需求冲突和不一致问题,以保证复杂系统的需求的正确性和一致性。另外,软件系统的非功能需求,如系统完整性和安全性等软件特性的验证需要通过执行软件来给予验证。但在需求分析阶段软件尚未开发出来,故要验证某些软件特性是相当困难的工作。若能在需求阶段通过分析目标系统的需求信息,严格地建立基于软件行为的需求模型,进而通过该模型刻画、验证相关的软件特性将是非常现实和重要的工作。因此,我们在行为模型的基础上提出了一系列语法检查方法,并且对系统的特性进行了逻辑刻画和检测,以验证行为模型的正确性和一致性。

基于上述研究,开发出了面向行为的需求建模工具 BRMT 实现了我们的建模及检测方法。图 1 所示为 BRMT 的主界面,BRMT 通过可视化操作界面来辅助开发人员完成建模以及检测等工作。

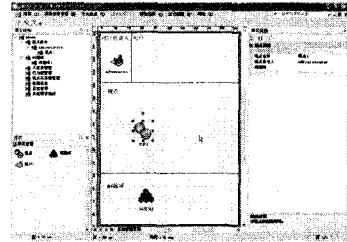


图 1 BRMT 主界面

本文第 2 节介绍基本概念;第 3 节阐述建模系统的结构;第 4 节结合一个实例详细介绍行为建模过程;第 5 节陈述基于 BDL 的需求验证;最后对全文进行了总结并阐述下一步的工作。

## 2 基本概念

本节介绍面向行为的需求建模方法中的一些基本概念。

### 2.1 问题域

所谓问题域是指与问题相关的部分现实世界。问题域包括所有与描述期望效果有关的事物,同时可用来产生这些效果的方法也是问题域的一部分。

对于复杂系统,其问题域中的事物繁多,事物间的关系错综复杂,很难直接进行建模。另一方面,项目相关人员的需求特别是功能方面的需求大多数仅与复杂问题域中的一部分相关。因此,一个较为可行的方法是首先将复杂系统分解为多个子问题域,然后在每个子问题域内分别进行建模。

### 2.2 视点

大规模和复杂的软件系统涉及到许多相关人员,这些人员由于各自的背景知识、职责及出发点不同,因此会从不同的角度和立场来提出自己的观点与需求,从而形成不同的需求。为此,在子问题域内建立多个视点,分别表示这些需求。一个观察者(视点源)根据其关注点和某个子问题域而提出的需求信息的集合构成一个视点。我们以视点模板的形式组织视点信息,视点模板由 9 个信息槽构成,每个信息槽记录视点的某方面的信息,如图 2 所示。

基本属性	
创建时间	2009-8-5 12:32:02
视点名称	视点1
视点责任人	user1
问题域	
最近修改时间	2009-8-3 12:30:51
需求属性	
关注点	
视点行为表达式	
视点源	
需求描述	

图 2 视点模板

视点模板中的创建时间和最近修改时间由系统自动生成。视点名称和视点责任人由项目负责人创建视点时指定,视点责任人是为该视点建立需求模型开发人员。问题域为该视点模型所描述的问题子域。关注点是视点在其所属问题域中所关注的方面。视点行为表达式是该视点模型的形式化描述。视点源描述视点中需求的来源,视点源可以为入、其它软件系统、外部硬件设备等。需求描述信息槽内填写视点的自然语言描述的原始需求。

### 2.3 场景

为了简洁独立地描述目标系统的各个执行过程,在视点需求开发的过程中,需要将视点分解成多个场景。具体需求内容的描述是在场景中进行的。场景指目标系统在某一执行期间内按顺序出现的一系列行为。场景的作用主要被用于描述该视点内,用户(或其他外部设备)与软件系统之间的一个或多个典型的交互过程,以便对软件系统需求中的行为有更具体的认识。

### 2.4 软件行为

软件行为是指软件运行时作为主体,依照自身的功能对客体进行施用、操作或动作的过程,或主体施用一个服务、操作或动作于客体<sup>[14]</sup>。软件行为由行为主体和客体、操作或动作、行为输入/出和行为属性等组成。

软件行为可分为原子行为(亦可称动作)或复杂行为(由多个原子行为或复杂行为构成)。行为与行为之间亦可进行通讯和交互。不同行为的区别主要体现在行为的主体、客体和操作(或动作)这三者之间有所不同。行为主体可以是用户或者问题域中的实体或概念。行为主体可以是复合主体,即一个行为可以有多个主体,且行为的主体必须是确定的。行为客体即行为的受体,通常指人或问题域中的实体或概念等。行为客体可以是复合客体,并且也可以是未确定的。

在行为描述语言 BDL 中,一个原子行为可表示为:

ABehID:  $f(SUB, OBJ[&obj \text{ 的补充说明}])$

[WHEN 前置条件]

[INFROM ( $ID_i$ )( $u_1, \dots, u_n$ )]

[OUTTO ( $ID_o$ )( $v_1, \dots, v_m$ )]

(1)

ABehID: IDLE

(2)

ABehID: RETURN([ABehID<sub>a</sub>])

(3)

式(1)表示当 WHEN 前置条件满足时,ABehID 行为的主体 SUB 对客体 OBJ 做了一个  $f$  动作,该行为从  $ID_i$  中接收  $n$  个参数  $u_1, \dots, u_n$ , 并且发送  $m$  个参数  $v_1, \dots, v_m$  到  $ID_o$  中。式(2)表示一个空行为,式(3)表示一个转移行为,若 RETURN 动作参数为空则结束;否则,跳转到指定的原子动作。

复合行为由多个行为组合而成,可由以下式子表达:

$B = ABehID \mid B_1; B_2 \mid B_1 \parallel B_2 \mid B_1 + B_2$

IF  $b$  THEN  $B_1$  ELSE  $B_2$  Fi

(4)

式(4)描述了 BDL 中的复合行为结构。复合行为可以是一个简单的原子行为,也可以是其它复合行为依照顺序、并行、非确定性选择、确定性选择这几种方式进行合成。

为描述复合行为的意义,我们使用结构化操作语义给出 BDL 复合行为的动态语义,这些规则是对 BDL 进行模型检测的基础。假设 ABehID,  $\alpha$  表示原子行为,  $B, B', B_i$  等表示复合行为。

(1)顺序

$\alpha; B \xrightarrow{\alpha} B$

(2)并行

$\frac{B_1 \xrightarrow{\alpha} B_1'}{B_1 \parallel B_2 \xrightarrow{\alpha} B_1' \parallel B_2} \quad \frac{B_2 \xrightarrow{\alpha} B_2'}{B_1 \parallel B_2 \xrightarrow{\alpha} B_1 \parallel B_2'}$

(3)非确定性选择

$\frac{B_1 \xrightarrow{\alpha} B_1'}{B_1 + B_2 \xrightarrow{\alpha} B_1'} \quad \frac{B_2 \xrightarrow{\alpha} B_2'}{B_1 + B_2 \xrightarrow{\alpha} B_2'}$

(4)确定性选择

$\frac{B_1 \xrightarrow{\alpha} B_1', b \text{ 为 true}}{\text{If } b \text{ Then } B_1 \text{ Else } B_2 \text{ Fi} \rightarrow B_1'}$

$\frac{B_2 \xrightarrow{\alpha} B_2', b \text{ 为 false}}{\text{If } b \text{ Then } B_1 \text{ Else } B_2 \text{ Fi} \rightarrow B_2'}$

## 3 建模系统结构

在软件需求过程中有很多繁琐复杂的工作,如怎样将用户需求建立成完整正确的需求模型,怎样以工程化的方式管理需求过程,怎样对需求模型进行验证等都是需要解决的问题。为此,面向行为的需求建模方法系统提出并实现了一套详尽的解决方案,以清晰的层次划分这些工作,以可视化的方式辅助开发者完成这些繁琐的工作,为开发者提供了方便快捷的需求建模手段。图 3 所示为 BRMT 的系统功能结构图。

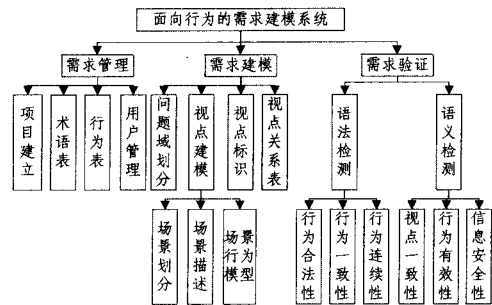


图 3 BRMT 结构图

系统结构主要划分为 3 大功能模块:需求管理、需求建模以及需求验证。

需求管理主要负责项目的管理、开发人员的管理以及需求模型中涉及到的术语表以及行为表的管理。

需求建模功能模块是 BRMT 的核心部分,其实现了以问题域划分为前提,以视点为建模的逻辑基础,以场景为需求模型建立的最小单位的需求建模方法。通过这些不同粒度的划分,可使复杂庞大的软件系统分解为层次清晰,逻辑相对独立的多个子模块,从而使建模问题得到简化。另外,本建模方法从软件行为角度着手,提取需求中的行为,对这些行为进行描述,并进一步对它们进行形式化建模,使用户的需求能更加清晰明确地表现出来。

需求验证工作是基于 BDL 形式化描述以及多视点划分来进行的。验证工作在基于 BDL 的软件行为模型基础上,以视点为最小单位,对行为模型进行语法、语义以及系统特性的检测,辅助开发人员对模型的正确性进行检验和更正。

为辅助需求开发小组的管理,系统提供了 AU(Administrator-User)两层用户管理模式,即为开发小组成员建立两个级别的账户:项目负责人和用户。一个项目有且仅有一个项目负责人,由项目负责人建立一个或多个用户账号。项目负责人负责全权负责项目的初始化工作,如项目的建立、问题域建立、视点建立、给视点指定负责人等,并且还负责建模过程中的一些全局的建模工作,如项目文本集成、维护视点关系表等;用户则以视点责任人的身份负责视观点级别的局部需求开发工作,这些工作包括给视点划分场景、建立场景行为模型、编辑视点行为表达式以及视点模型建立后的一系列语法检测工作。

## 4 面向行为的需求建模过程

使用 BRMT 对目标系统进行需求建模。首先对目标系统进行自顶向下的分解,然后用行为描述语言自底向上地为目标系统建立行为模型。具体的建模过程如图 4 所示。

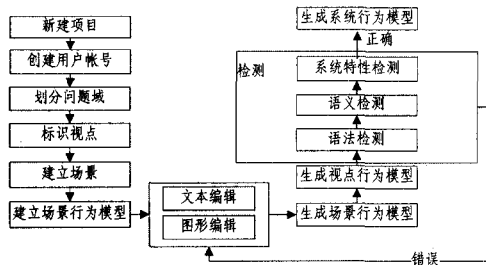


图 4 建模步骤

首先将目标系统划分成多个子题域,然后在每个子题域中标识视点,用以管理需求相关信息及需求模型。每个视点下可建立多个场景和场景行为模型,需求信息转换成形式化的行为描述语言,以场景行为模型为最小单位存放。视点内所有场景的行为模型组合构成视点行为模型,目标系统所有视点的行为模型组合构成系统行为模型。

下面结合一个手机金融系统的实例对需求建模步骤进行介绍。

### 4.1 实例描述

某型手机金融系统是股市行情在手机上的实现,使用户可以通过某型手机了解股市和外汇等的实时行情。在股市实时行情列表中,提供了所有股票和外汇的实时信息,并提供动态更新功能;同时,提供技术分析图、多项技术指标图,便于用户决策。用户还可以搜索相应股票信息。

整个系统由两个部分组成:服务端和客户端。

服务端的主要功能在于:

- (1)与数据源连接,采集实时的股市行情和外汇数据,新闻信息的采集;
- (2)发送最新信息数据到客户端;
- (3)进行用户的帐号管理和新闻类的管理,此功能在第二期实现;
- (4)数据存储,并生成分段数据。

客户端的主要功能在于:

- (1)在界面上显示实时行情图;
- (2)提供技术分析图,以便于用户分析股市行情。

### 4.2 划分问题域

首先对系统的问题域进行划分,问题域划分的具体步骤如下:

- (1)开发人员与客户、应用(问题)领域专家等对问题进行交流;
- (2)确定待开发系统问题域的边界;
- (3)将系统按照松耦合高内聚的原则将待开发系统的问题域划分为多个子问题域;
- (4)项目负责人登录系统,在系统中新建问题子域。

容易发现,本系统包含三个层次,第一层是系统的直接客户,他们与手机系统进行交互;第二层是手机上的终端系统,负责为第一层的客户提供交互界面,并接收用户的各种请求

和输入,并将数据传输至主机;第三层是本系统的服务器部分,它负责存储、处理各种数据及响应终端系统的各种请求,并将处理结果返回给终端系统。因此,根据手机金融系统的层次关系,可将其划分为两个问题域:

服务器端问题域:由服务器、数据库、数据采集端口等构成;

客户端问题域:手机客户端系统。

### 4.3 标识视点

视点是需求本建模方法中建模的逻辑单位,项目负责人根据相应的问题子域,从中找出所有的视点源及其关注点,将它们标识为视点,并指定视点间关系。为每个视点建立一个视点对象。并且给每个视点制定一个责任人,该视点的后续建模工作由该责任人负责完成。标识视点的具体步骤如下。

对于每个问题域:

- (1)分析并确定出该问题子域中存在的需求源即视点源;
- (2)确定每个视点源对问题子域的关注点;
- (3)根据关注点创建视点,生成视点模板;
- (4)填写视点基本信息,如视点标识、视点责任人等。

根据上面的步骤,我们将手机金融系统标识出如下几个视点:

(1)对于服务器端问题域

- a)数据管理视点;
- b)实时行情数据管理视点;
- c)新闻信息管理视点;
- d)数据请求管理视点;
- e)用户管理视点。

(2)对于客户端问题域

- a)软件获取视点;
- b)业务逻辑处理视点;
- c)行情展示视点;
- d)走势分析视点。

### 4.4 建立场景

视点责任人进入视点编辑界面后,根据待开发软件系统交互情况,将视点划分成多个场景。

建立场景的过程分为如下几步:

(1)确定系统的所有行为和主体

分析视点中用自然语言描述的需求,并从中归纳出描述场景的动作和主客体。然后,将动作及相应的主客体一起视为软件系统中的行为,且主客体分别被视为行为的主体和客体。此处的行为主体不仅包括使用系统的用户,而且也包括一切参与到行为中的实体和概念等。

(2)确定有效的行为和主客体

在获得需求中描述的动作和行为主客体之后,进一步对其进行分析和筛选,去掉不正确和不必要的行为和行为主客体,保留那些与系统密切相关的行为和行为主客体。

(3)分析行为间的关系

行为间的执行关系大体上可以分为顺序、并行、交互、选择以及屏蔽等,行为在结构上也可分为父行为与子行为之间的关系。在自然语言描述的需求中,通过分析描述性动词或动词词组之间的关系,可以推导出行为间的关系。同时,通过分析,还能发现一些在自然语言描述的需求中隐含的行为间

的关系。

#### (4) 构建场景

场景的描述主要用自然语言,而且根据已获取的所有有效行为并按它们的执行顺序构成有序的行为系列,从而构建出每一个场景。

下面以“新闻信息管理视点”为例介绍场景的建立。

新闻信息管理视点的需求描述:新闻的管理包括对数据库中的新闻的增、删、查操作,新闻的增加数据表示新闻的采集,新闻采集有两个方式:通过 HTTP 的方式从网站采集和从文本中采集,并自动添加至数据库中。同时,还提供将新闻广播 push 的功能,将相关新闻 push 到手机上去。由于涉及到数据采集源的问题,本期先不考虑新闻的采集及 push 处理。

我们为视点建立 3 个场景:新闻采集(HTTP 方式采集)、新闻采集(文本采集)、新闻 push。3 个场景的内容描述如下。

##### (1)新闻采集(HTTP 方式采集)

功能说明:通过 HTTP 协议采集新闻。

用例描述:服务端通过向某网站发出 HTTP 请求采集到新闻,并存放至数据库中。

行为序列描述:

前置条件:要访问的 url 提供了新闻。

1) 服务端定时访问某新闻站点的新闻 url,获取其新闻内容;

2) 新闻站点返回新闻内容;

3) 服务端解析其 RSS 格式的新闻内容;

4) 将获取的新闻存入新闻数据库中。

##### (2)新闻采集(文本采集)

功能说明:从文本中采集新闻。

用例描述:服务端通过获取的文本中采集到新闻,并存放至数据库中。

行为序列描述:

前置条件:获取了新闻文本。

1)服务端解析获取的新闻文本;

2)将解析好的新闻存入数据库。

##### (3)新闻 push

功能说明:将相关的新闻 push 到用户。

用例描述:管理者在界面上查看采集得到的新闻,如果需要将此新闻广播出去,则按页面上的 push 按钮,将此新闻发送到所有正在运行的用户中。

行为序列描述:

前置条件:服务端有新闻信息。

1)管理者通过点击新闻页面上的“push”按钮,将显示的新闻广播出去;

2)服务端向数据库中获取当前所有在线用户的 pin 码;

3)服务端将此新闻信息通过 pin 码 push 给所有的用户。

#### 4.5 建立场景行为模型 SCBM

对于每一个场景,我们用 BDL 为场景建立对应的 SCBM,对场景内的需求进行形式化的描述。每个 SCBM 由一个行为序列组成,一个 SCBM 为:

$$SCBM=(B,+ ,If,||,; ) \quad (5)$$

式中, $B$  表示场景内所有行为(行为表达式)的集合。 $+$ , $If$ , $||$ , $;$ 分别表示  $B$  中行为间的非确定和确定选择,并行和顺序关系。

SCBM 的 BDL 结构如图 5 所示。

场景 ID [ $]$ ,场景 ID]:(符号“ $[ ]$ ”表示其中的内容是可选的)

BEGIN

[ ABEH:

ABehID:原子行为;

.....

ABehID:原子行为 $n$ ](注:原子行为列表)

BEH:

BehID=场景行为表达式

[BehID=子行为表达式 $i$ ]

.....

[BehID=子行为表达式 $m$ ]

END(注:允许将同一视点中多个场景合写为一个行为表达式)

图 5 场景行为模型

BRMT 提供了两种 SCBM 的编辑模式:文本编辑模式和图形编辑模式。在文本编辑模式下,开发人员可通过直接书写行为表达式进行 SCBM 的 BDL 代码编辑,也可以在图形编辑模式下通过场景行为模型图(SCenario Behavior Model Diagram SCBMD)来表达场景,SCBMD 以格的结构与 SCBM 结构相对应,有关 SCBMD 的内容将在后文中予以介绍。

#### 4.6 建立视点行为模型 VBM

视点的需求模型由 VBM 来表示。一个 VBM 为:

$$VBM=(S,+ ,If,||,; ) \quad (5)$$

式中, $S$  表示视点内所有场景的集合,且  $S$  中每个场景对应一个场景行为模型。 $+$ , $If$ , $||$ , $;$ 分别表示  $S$  中场景间的非确定和确定选择,并行和顺序关系。VBM 的 BDL 结构如图 6 所示。

视点 ID:

VPBEGIN

场景 ID $i$  的行为模型

.....

场景 ID $n$  的行为模型

VPBehID =视点行为表达式

=场景的 BehID 场景间关系符 场景的 BehID

[场景间关系符 场景的 BehID .....]

VPEND

图 6 视点行为模型

图 7 所示为 BDL 描述的新闻信息管理视点的 VBM。

VPNewsDataMan://新闻信息管理视点

VPBEGIN //视点描述开始

//新闻采集(HTTP 方式)场景

BEGIN //场景描述开始

ABEH //原子行为

AccessNewsURL:访问(服务器,新闻 URL)

ColNewsContent:获取(服务器,RSS 格式的新闻)

AnalyzeRSS:解析(服务器,RSS 格式的新闻)

StoreNews:存储(服务器,解析后的新闻)

Return31;RETURN()

BEH //复合行为

//新闻采集(HTTP 方式)场景行为表达式

```

BehColNewsHttp=
AccessNewsURL;
ColNewsContent;
AnalyzeRSS;
StoreNews;
Return31. //退出系统
END //新闻采集(HTTP方式)场景描述结束
//新闻采集(文本采集)场景
BEGIN //场景描述开始
ABEH //原子行为
    AnalyzeRSS1:解析(服务器,新闻文本)
    StoreNews1:存储(服务器,解析后的新闻)
    Return32;RETURN()
BEH //复合行为
//新闻采集(文本采集)场景行为表达式
BehColNewsText=
    AnalyzeRSS1;
    StoreNews1;
    Return32. //退出系统
END //新闻采集(文本采集)场景描述结束
//新闻 push 场景
BEGIN //场景描述开始
ABEH //原子行为
    ClickPush:点击(管理员, push 按钮)
    GetCustomPin:获取(服务器, pin 码)
        INFROM(! 数据库)(pin 码)
    PushNews:Push(服务器,新闻)
        OUTTO(! 客户端)(新闻)
    Return33;RETURN()
BEH //复合行为
//新闻push 场景行为表达式
BehPushNews=
    ClickPush;
    GetCustomPin;
    PushNews;
    Return33. //退出系统
END //新闻 push 场景描述结束
VPNewsDataMan =
    (BehColNewsHttp|BehColNewsText);BehPushNews
//新闻信息管理视点的行为表达式
VPEND

```

图7 视点行为模型实例

#### 4.7 建立系统行为模型 SBM

综合所有视点的行为模型,得到目标软件系统最终的需求模型。一个系统的行为模型为:

$$SBM=(V, R_0, R_1, R_2)$$

式中,  $V$  表示与系统相关的视点的集合,且  $V$  中每个视点对应一个视点行为模型。 $R_0, R_1, R_2$  分别表示  $V$  中视点间的重叠、顺序和无关关系,视点之间的关系在视点关系表中表示。

建立的系统行为模型如图8所示。

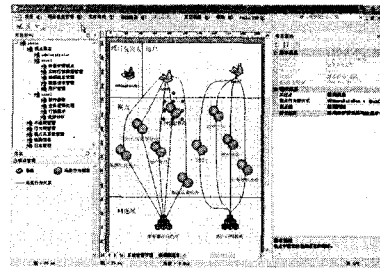


图8 某手机金融系统行为需求模型

## 5 需求验证

我们使用模型检测的方式对行为模型的正确性进行验证,所做的检测分为两个方面:

语法检测:行为表达合法性、行为一致性和行为连续性;

语义检测:视点一致性、行为有效性、信息安全性。

### (1)行为合法性检测

行为合法性检测是以视点为单位检查其中行为模型的语法合法性。

给定一个视点的行为模型(或行为表达式集合),称其表达是合法的,如果该行为模型(或行为表达式集合)中所有的行为表达式能满足行为描述语言的语法。

### (2)行为一致性检测

行为模型通过行为表达合法性检测后,需要检测行为模型的静态一致性。静态一致性包括视点内的一致性和视点间的一致性。

视点内的行为一致性:给定一个视点行为模型,称其中含输入/出的行为是一致的,如果该视点行为模型中的某些含输出的行为都存在一个与其匹配的含输入的行为,且该行为在本视点模型中。

视点间的行为一致性:假设系统行为模型中存在两个视点  $V_1$  与  $V_2$ ,称视点  $V_1$  与  $V_2$  中含输入/出的行为是一致的,如果  $V_1$  的行为模型中存在一个与  $V_2$  相关的带输出参数的行为,并且  $V_2$  的行为模型中至少存在一个含输入参数的行为与其匹配(即数据名相同)。或者,反之也成立。

### (3)行为连续性检测

行为连续性是指在一个行为模型中行为与行为之间至少存在一条执行路径,即行为间存在可达关系。

称一个行为模型是连续的,如果在与该行为模型对应的行为树中,所有定义的行为都是可达的。行为模型的连续性表示该行为模型是可以运行的。

### (4)视点一致性检测

视点间行为一致性是指在一个由多个视点构成的系统行为模型中,如果其中两个视点出现重叠,则对于重叠部分中相同行为的理解和描述应该是相同的。

对于两个行为  $f(sub_1, obj_1)$  和  $g(sub_2, obj_2)$ ,称它们是相同的,当且仅当  $f=g, sub_1=sub_2, obj_1=obj_2$ ,而且两个行为中出现的输入和输出也相同。

给定两个重叠的视点  $V_1$  和  $V_2$ ,令  $(V_1, V_2)$  为这两个视点重叠部分中相同行为的集合,称  $V_1$  与  $V_2$  是一致的,当且仅当与视点  $V_1$  和  $V_2$  对应的行为模型  $M_1$  和  $M_2$  在  $(V_1, V_2)$  上是观察等价的。

### (5)行为有效性

行为有效性表示一个行为模型总能够按照预期的方式运行。这里指的预期的方式就是一组行为踪迹的集合,也可以用时序逻辑公式表示。称一个行为模型  $M$  是有效的,如果该模型能满足所有预期的运行踪迹(时序逻辑公式 4 成立)。我们定义了 4 种特性:

**系统一致性:**系统一致性描述行为模型中的行为的执行无矛盾性及无二义性,即系统不希望发生的行为一定不会发生且期望的行为肯定发生。

**系统安全性:**在一定条件下,某些不希望的行为永远不会发生。

**行为可信性:**在一定条件下某些期待的行为终究会发生。

**行为非终止性:**在一定条件下,某些行为会无限经常地发生。

我们用时序逻辑公式对这些特性进行了刻画,并在该基础上进行检测。开发人员可根据需要,使用逻辑公式定义更多的系统特性,在检测工具中进行检测。图 9 所示为有效性检测界面。

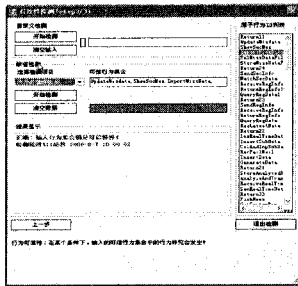


图 9 有效性检测

### (6)信息安全性检测

系统信息安全性包含机密性、完整性与可用性。其中,反映机密性与完整性本质的原始模型是  $BLP^{[15]}$  和  $Biba^{[16]}$  模型。由于完整性与机密性是互为对偶的,因此,这里讨论并实现完整性的检测功能。

系统的复合行为能够最终分解为原子行为。如果经过检测,所有的原子行为均符合  $Biba$  模型的严格完整性策略(SI),则称整个系统的安全性(完整性)得到了保证。

**结束语** 本文介绍了面向行为的需求建模方法,并且实现了可视化的需求建模工具以辅助分析人员使用本方法进行建模和验证工作。对于复杂的系统,首先将目标系统问题域划分成多个子问题域,然后在子问题域中建立多个视点,每个视点内又可建立多个场景。根据用户的需求内容,在场景内用 BDL 描述软件执行序列,以建立场景行为模型。一个视点内的场景行为表达式有可能组合成视点行为模型。最后,目标系统内所有的视点行为表达式构成系统的行为模型。为验证行为模型的正确性,我们提出了一套语法、语义以及特性的检测。基于上述方法,我们开发了面向行为的需求建模工具 BRMT,以辅助开发人员从行为的角度进行需求建模。

面向行为的需求建模方法通过子问题域、视点和场景,对目标系统进行自顶向下的分解,然后从每个场景开始,自底向上地建立行为模型。以视点为单位给分析员分配建模任务,以场景为最小单位进行行为建模,能够有效地组织和管理开发过程。

BDL 直接描述目标系统的行为,原子行为的结构接近自

然语言的表达方式,可读性强,易于掌握、使用和理解。复合行为以一致的形式对不同粒度的行为进行描述,应用灵活。我们基于 BDL 基本语法研究出了一套图形表示方法 SBMD,辅助开发人员快捷直观地建立行为模型。我们给 BDL 建立了严格的语义,并在此基础上提出了一套验证方法对行为模型进行验证,辅助开发人员检验行为模型的正确性。面向行为的需求建模方法是基于形式化描述的建模方法,但大部分操作都是可视化的,有效促进了形式化方法的普及应用。

我们已经使用 BRMT 对多个系统实例进行了需求建模,取得了良好的效果,但仍有很多方面需要改进和完善。BDL 能够很好地表达软件行为的执行,不过仍需要通过更多的实例,验证 BDL 的表达能力。我们在 BDL 模型上刻画并检测了一些系统特性,更多的检测方法和技术还有待进一步的研究和实现。

## 参 考 文 献

- [1] The Standish Group. CHAOS Summary 2009[R]. Technical the Standish Group International, 2009
- [2] The Standish Group. Report of the Standish Group International [R]. Technical The Standish Group International, 1998
- [3] Milner R. A Calculus of Communicating Systems[M]. Springer-Verlag, 1982
- [4] Brookes S D, Hoare C R A, Roscoe A W. A Theory of Communicating Sequential Processes[J]. Journal of the ACM(JACM), 1984, 31(3): 560-599
- [5] Pnueli A. The temporal logic of programs[C]//Proceedings of 18th Annual Symposium on Foundations of Computer Science. 1977:46-57
- [6] Murata T. Petri nets: Properties, analysis and applications[J]. Proceedings of the IEEE, 1989, 77(4): 541-580
- [7] OMG. UML super structure, version 2.0 [R]. Object Management Group, 2005
- [8] 陆汝钐, 金芝. 面向本体的需求分析[J]. 软件学报, 2000, 11(008): 1009-1017
- [9] 吴步丹, 金芝, 赵彬. 面向服务的建模: 一种全过程复用的方法[J]. 计算机学报, 2008, 31(008): 1293-1308
- [10] 张伟, 梅宏. 一种面向特征的领域模型及其建模过程[J]. 软件学报, 2003, 14(8)
- [11] Wan Li, Wu Guoqing, Wu Huaiguang. BDL-Behavior Description Language[C]//Mahadevan V, Zhou J H, Ng A. proceeding of 2009 International Conference on Software Technology and Engineering. SINGAPORE: World Scientific, 2009: 37-41
- [12] Mullery G P. CORE-a method for controlled requirement specification [C] // Software engineering. Munich, Germany: IEEE Press, 1979: 126-135
- [13] Nuseibeh B, Kramer J, Finkelstein A. A Framework for Expressing the Relationships Between Multiple Views in Requirements Specification[J]. IEEE Transactions on Software Engineering, 1994, 20(10): 760-773
- [14] 屈延文. 软件行为学[M]. 北京: 电子工业出版社, 2004
- [15] Bell D E, Lapadula L J. Secure Computer Systems: Mathematical Foundations and Model[M]. Bedford, MA, USA: The MITRE Corporation, 1974: M74-M244
- [16] Biba K J. Integrity Considerations for Secure Computer Systems [R]. Mitre Corp. Report TR-3153. Bedford, Mass, 1977