

基于规则的用例规约验证方法

张莹¹ 吴际¹ 刘超¹ 杨海燕¹ 胡宁²

(北京航空航天大学计算机科学与技术系 北京 100191)¹

(中航工业西安航空计算技术研究所 西安 710065)²

摘要 用例模型描述了参与者对系统功能的需求,在整个系统的开发过程中有着重要作用;用例描述若存在问题,会对系统开发造成很大影响。提出了一种用例规约的规则验证方法,通过分析用例规约中的错误类别,在特定的用例规约描述方法上设计了帮助发现规约中不完整性、不一致性错误的验证规则,并通过规则的形式化来支持需求的自动化验证。

关键词 需求验证,验证规则,需求错误,限制的用例规约建模方法

中图法分类号 TP311 文献标识码 A DOI 10.11896/j.issn.1002-137X.2017.04.026

Rule-based Verification of Use Case Specification

ZHANG Ying¹ WU Ji¹ LIU Chao¹ YANG Hai-yan¹ HU Ning²

(Department of Computer Science and Technology, Beihang University, Beijing 100191, China)¹

(Xi'an Aeronautics Computing Technique Research Institute, AVIC, Xi'an 710065, China)²

Abstract Use cases capture the functional requirement of system and are important in software development. Any error in use cases may have influence on the development of system. This paper described a method to verify use case specification based on the rules. First, we analyzed the error types of use case specification, then proposed a set of rules to help find these errors, which supports the automatic verification by formalizing the rules and focused the verification work on the meta-model of a kind of use case modeling method called RUCM(Restricted Use Case Modeling).

Keywords Requirement verification, Verification rules, Requirement error, RUCM

1 前言

在软件开发活动中,需求工程是最重要、最关键的活动。为了提高需求规约的质量,研究人员在检测、消除需求缺陷的方法上做了很多尝试。在对工业领域的需求调研^[1]时发现,需求中最常见的缺陷有4类:1)需求通常被表达为解决方案;2)二义性;3)不完整;4)不一致。在对需求缺陷的研究^[2]中将需求分为冗余的、不完备的、不正确的、不一致的和其它问题这5类,也强调了需求中的不完整和不一致问题。因此,需求中的不完整与不一致错误应该受到重视。另外,为了支持需求的分析与验证,提高需求的易理解性,对于采用特定需求描述方法的需求,应检测其是否满足描述方法的规范,例如,在对需求描述的预定义语言模式^[3]的研究中提到了应对需求的模式进行匹配检查。在基于规则的场景验证方法中^[4],由于采用了特定的场景描述语言,在验证前要检测描述中的名词类型使用错误和范例缺失的错误。因此,需求是否按照描述方法约定进行描述也是验证中应该关注的问题。

用例模型包括用例图与用例规约,用例规约描述了用例

的使用上下文和系统与参与者的交互动作以及系统对用户输入的处理。为了支持用例的自动化验证,通常采用形式化(如Z语言)或在验证中补充信息等方式来描述需求。本文通过分析需求、用例规约中的不完整和不一致以及不符合需求描述方法约定这3个方面的错误,借助RUCM用例建模工具能够自动生成需求的形式化表达的优点,提出验证规则来支持用例规约的自动化验证,帮助发现用例规约中的错误。

本文第2节介绍用例模型中的错误类别,通过分析概念,结合用例需求的特点来分析用例规约中的错误类别;第3节针对特定的用例建模方法RUCM,分析错误类型并且设计规则;第4节给出案例分析,通过真实案例说明该方法的有效性;第5节介绍相关研究;最后总结全文。

2 用例规约中的需求错误分析

2.1 需求不完整错误

一般而言,需求完整^[5]意味着:1)需求包括了软件应该实现的所有功能;2)需求为所有有效的输入定义了响应;3)需求描述涉及的图表都有完整标识和描述;4)需求描述中涉及的

到稿日期:2015-11-30 返修日期:2016-02-27 本文受国家重点科研项目(MJ-S-2012-05)资助。

张莹(1992—),女,硕士,主要研究领域为需求建模、需求验证,E-mail:480zhangying@163.com;吴际(1974—),男,副教授,主要研究领域为软件安全性与可靠性、嵌入式软件设计与验证、软件测试;刘超(1958—),男,博士,教授,主要研究领域为软件测试、面向对象技术、软件开发环境等;杨海燕(1974—),女,硕士,讲师,主要研究领域为软件工程;胡宁(1978—),男,博士,高级工程师,主要研究领域为操作系统和软件工程。

计量术语和单位有完整定义。由于不同软件的需求描述内容和采用的描述方式存在差异,因此在需求确认活动中对完整性的确认有所差异。需求确认确保需求描述了系统利益相关者对系统的所有功能要求;需求验证依据现有需求,结合需求描述方法,从需求本身逻辑中找到缺失的内容。例如,在使用用例的需求描述中,要求每个用例和动作都有前置条件和后置条件^[6];在涉及资源描述的用例需求中,要求每个资源都应该在用例的场景描述中出现^[7]。

本文针对基于用例的需求描述,重点考虑上述 1) 和 2) 两方面的完整性要求。用例规约强调描述参与者与软件的交互,软件的输入就是参与者发出的动作和相应数据,软件对输入的响应则是软件采取的执行动作。基于这样的理解,用例规约中经常存在的不完整缺陷主要包括:1)参与者未定义,即:用例的参与者没有在需求中进行定义^[8];2)缺失的动作,即:用例规约中参与者与系统的交互流中有缺失的参与者动作或系统动作,例如安全关键软件需求中对特定异常的系统处理动作缺失,系统使用了外部参与者的数据,但是并没有对应的数据采集或数据输入动作,所描述的事件流动作不足以完成用例的目标^[9];3)特定需求描述方法规定的描述内容缺失,特定的用例规约描述方法在描述中可能会引入特定的概念和内容,一旦缺失也会导致需求描述不完整,如文献^[7]在用例规约中要求描述用例需要的资源,因此在完整性规则中提出需要验证用例资源的定义是否完整。

2.2 需求不一致错误

需求一致性指需求描述内部是否有逻辑矛盾的特性^[5],如果有逻辑矛盾,则意味着需求不一致,否则为需求一致,任何需求建模方法都要有自己的一套规则来检查需求模型的一致性^[10]。需求不一致问题主要表现为^[11]:对外部对象(如资源或者外部实体)的要求或约束不一致;需求描述动作之间在描述逻辑或执行时序上不一致;对一个对象的描述和术语使用不一致。

软件在执行功能的过程中往往涉及到与外部对象的交互,为了保证软件能够达到预期的效果,往往需要对外部对象的状态做出相应的要求或约束。所谓外部对象的要求或约束不一致,是指不同需求或者同一需求内部施加于同一外部对象的约束或要求不一致。例如,某直升机的最大俯仰角为±10°,俯仰角在需求 1 中的划分为俯仰角>10°或俯仰角<-10°,10°≤俯仰角≤10°,而在需求 2 中的划分为俯仰角>20°或俯仰角<-20°,20°≤俯仰角≤20°,这就说明对数据划分的要求存在不一致。

不同的需求有可能会涉及相同的处理场景片段,这时如果参与者提供的输入相同,需求所采取的处理也应该相同,从而使得用户能够获得一致的功能体验。所谓需求描述动作之间的逻辑或时序冲突,是指两个用例在相同的处理场景下^[8]针对同一个参与者的输入所做出的响应动作不一致。例如,在一个系统的不同功能中,都需要用户输入登录密码,系统进行验证。在功能 A 中指明用户密码输错 3 次后系统退出,而在功能 B 中声明输错 5 次密码后系统退出,属于需求中的不一致问题。

需求描述中常常使用一些命名来指代某些对象。所谓对

对象的描述或术语的使用不一致,是指使用不同的命名来指代相同的对象,导致需求理解困难。如在电梯系统需求中,电梯门控制计时器在用例图中使用 DoorTimer 来指代,而在用例规格中用 ElevatorTimer 来指代,存在不一致错误。

2.3 不符合需求描述规范的错误

用例规约一般采用自然语言描述,自然语言会带来二义性问题^[12],因此不同的需求描述方法都设计了相应的约束来规范自然语言的使用以减少产生二义性的机会。主要约束有 3 类^[13]:1)通过自然语言使用规则和关键字来限制^[14],这些方法主要定义需求中频繁使用的关键字、词组和句子结构;2)通过描述指导规则来限制^[15],这些方法总结了书写自然语言需求规约时的指导规则,比如主要子句的动词应该用主动语态;3)针对领域的语言使用模式^[16-17]。所谓不符合需求描述方法规范,是指没有按照描述方法进行描述,可能会使得需求不能很好地被理解,分析与验证时会出现错误。

用例规约需求错误分类如图 1 所示。

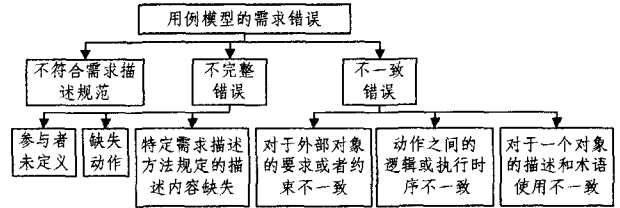


图 1 用例规约需求错误分类

3 面向 RUCM 的错误分类与验证规则设计

3.1 RUCM 需求建模方法

RUCM(Restricted Use Case Modeling)需求建模方法^[18]是一种基于用例的方法,由结构化的用例规约模板和一组限制规则构成。结构化的模板定义了在用例规约中需要描述的内容,如表 1 所列。限制规则包括自然语言规则与关键字。实验证明,RUCM 方法可以有效地降低需求规约的二义性,同时又保留自然语言易理解和易使用的优点。

表 1 RUCM 用例规约模板

Use Case Name	用例名称,通常以动词开头	
Brief Description	用例内容的简要描述	
Precondition	用例的前置条件,即用例执行前必须满足的条件	
Primary Actor	主要参与者	
Secondary Actors	次要参与者	
Dependency	依赖关系,描述与其他用例之间的包含、扩展关系	
Generalization	泛化关系,描述与其他用例之间的泛化关系	
Basic Flow	Steps	事件流步骤
	PostCondition	基本流的后置条件
	GuardField	引用流中的步骤号或守护条件
Alternative Flows	Steps	事件流步骤
	PostCondition	分支流的后置条件

北京航空航天大学已经针对 RUCM 方法开发出相应的建模工具^[19],该工具支持 RUCM 用例模板和约束规则,并能够对 RUCM 关键词及其用法进行检查。RUCM 建模工具能够将 RUCM 需求转化为 UCMeta 的实例模型^[20],从而支持对 RUCM 需求模型的自动化分析^[18]。RUCM 用例规约模板如表 1 所列。

3.2 UCMeta

UCMeta 定义了 RUCM 用例模型的所有必要概念及其

关系,由用例图、用例规约模板、自然语言元模型组成。用例规约模板元模型元素如图2所示。用例规约中一个重要的概

念是事件流(FlowOfEvent),它按时间次序描述了参与者和系统交互的具体步骤,由语句集合(steps)构成。

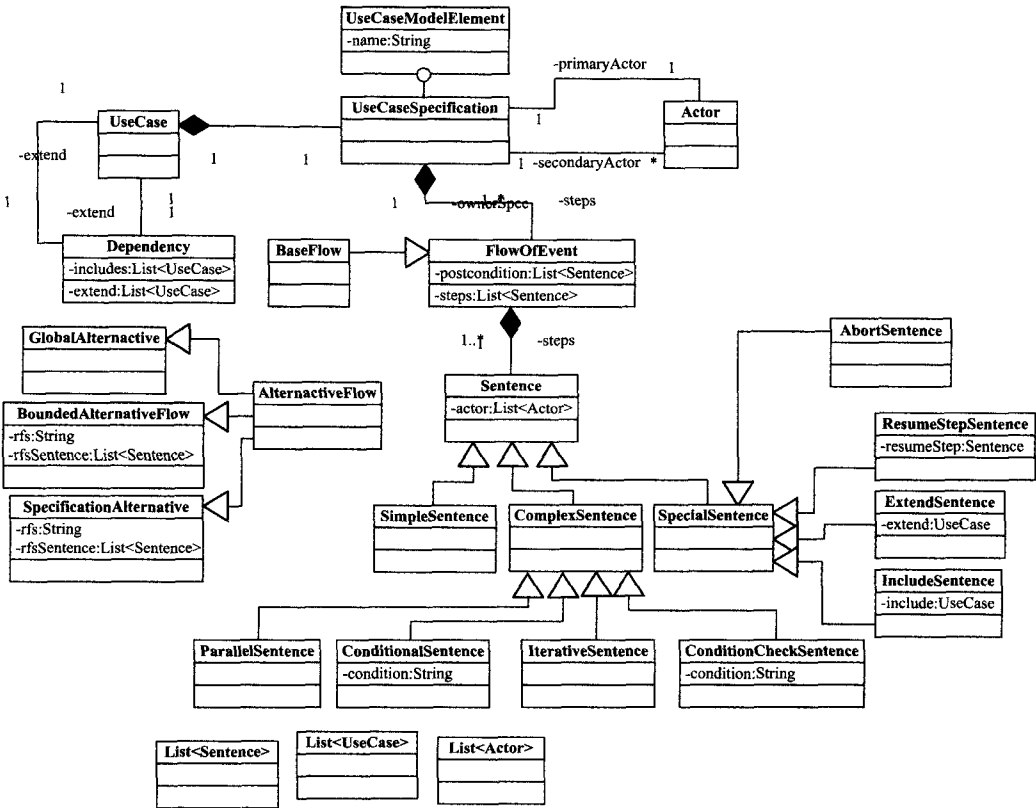


图2 RUCM元模型

3.3 规则的形式化定义

为了支持基于规则的自动化验证,有两点要求:1)形式化地描述规则,使得规则能够被计算机识别和分析;2)每条验证规则中的验证对象需要表示元模型中的元素。RUCM建模工具能够将需求自动形式化为部分UCMeta的实例模型,因此验证引擎能够从建模工具中直接取得验证对象进而进行验证。

本研究提供了规则描述的语法规则,BNF范式描述如下:

```

<VerificationRule> ::= <VerificationObject>
<keyword>[ReferenceObject]
<VerificationObject> ::= <UCMeta>{. <attribute>} { U
VerificationObject }
<ReferenceObject> ::= <UCMeta> { . <attribute>} { U
VerificationObject } | <Format>
<keyword> ::= IsSubsetOf | IsNotNull | ConformToFormat | isUnique

```

VerificationObject描述验证对象,ReferenceObject描述验证的参考对象,为UCMeta实例集合,集合可以进行并运算(U)。keyword表示验证规则的约束逻辑;IsSubsetOf表示满足集合属于关系;ConformToFormat表示满足格式要求;IsNotNull表示集合中每个对象值非空;isUnique表示集合元素唯一。规则所表示的对于验证对象集合的约束表达的是集合中的每个元素满足约束。另外,参考对象有一类为格式,目前RUCM方法中对格式有明确要求的为语句引用表达式。

3.4 针对描述规范符合性的错误检查

RUCM建模方法包括用例规约模板和限制规则,前者为需求描述提供了规范的模板,后者则针对不同类别的软件处理动作和自然语言使用提出了限制规则,包括两部分:描述特定动作或流程的关键词,例如支持控制流程描述的IF-THEN-ELSE-ELSEIF和DO-UNTIL关键词、表示系统验证动作的VALIDATES THAT关键词、表示用例之间关系的ICNLUDE USE CASE和EXTEND USE CASE关键词;限制自然语言使用的规则,比如事件流中句子的主语必须是“系统”或参与者,时态只能使用现在时。因此错误可以分为3类:1)模板中对内容的格式要求,自动化的分析与验证要求需求描述严格按照约定的格式,若不满足格式要求,则会影响自动化验证的结果;2)自然语言限制规则;3)关键字的使用正确。其中类型2)中的规则验证在RUCM的自然语言分析中进行,不在本方法研究范围内;类型3)的验证已经在RUCM建模工具中实现,即规约编辑器对RUCM关键词的用法检查。

目前模板中对内容格式的要求主要是针对引用语句表达式,其出现在两个位置:1)特定分支流与有界分支流的引用语句表达;2)ResumeStepSentence中的引用语句表达。

规则1 分支流中对语句的引用表达符合表达语法规范。

SpecificAltercative. rfs U BoundedAltercative. rfs U ResumeStepSentence. rfs ConformToFormat RFSFormat

3.5 不完整错误及其检查规则

3.5.1 参与者未定义

参与者未定义在 RUCM 用例规约中指的是参与者描述域(primaryActor 和 secondaryActor)中没有列出所有与该用例相关的参与者(UseCase. actor, 从自然语言分析结果中提取)。当事件流中出现了用例参与者中不存在的参与者时,认为参与者缺失。

规则 2 事件流中的参与者在用例的参与者描述域中存在。

Sentence. actor IsSubsetOf UseCase. actor

3.5.2 缺失动作

缺失动作指事件流中的动作缺失。针对 RUCM 方法所定义的句型和描述,本文将从条件验证语句的异常处理、分支流的结束动作和参与者在事件流中的参与 3 个方面来识别可能的问题。

ConditionCheckSentence 是 RUCM 中常用的一类条件验证语句,用于对外部对象或者系统内部状态进行检查,并针对两种可能结果(即满足条件和不满足条件)进行不同的处理。常见的问题是缺失对不满足条件的处理动作。

规则 3 条件判断语句有对应的处理分支流。

ConditionCheckSentence IsSubsetOf SpecificAlternative. rfsSentence ∪ BoundedAlternative. rfsSentence

分支流结束可能有两种情况:返回某个事件流或直接退出,若未指明返回或退出,由于没有默认的约定结束方式,因此认为事件流的逻辑不完整。对于返回某个事件流,RUCM 方法通过“事件流加步骤号”的方式指明返回事件流中的语句,因此要求引用语句存在才能保证返回动作是有意义的,若不存在,则可能是因为对应的语句由于需求的改变而变更了位置或编号书写错误。为了支持验证,通过 AlternativeFlow. lastSentece 表示分支流语句集合的最后一个语句。

规则 4 分支事件流以退出动作或返回动作结尾。

AlternativeFlow. lastSentence IsSubsetOf AbortSentence ∪ ResumeStepSentence

规则 5 返回动作语句中引用的语句存在。

ResumeStepSentence. resumeStep IsSubsetOf Sentence

用例参与者应该参与到用例事件流中某些动作中,否则可能认为是事件流中缺失了该参与者的动作或参与者定义多余。

规则 6 用例相关参与者在事件流中有相应的动作。

UseCase. actor IsSubsetOf Sentence. actor

3.5.3 特定需求描述方法规定的描述内容缺失

RUCM 方法规定每个事件流都必须有后置条件,明确流程在完成应该处于何种状态。在基于用例生成抽象测试用例^[21]时,后置条件是判断一个功能是否正确完成的依据。

规则 7 事件流的后置条件不能为空。

FlowOfEvent. postcondition IsNotNull

用例功能从主事件流开始,分支流只有在特定的条件下才会触发,若分支流没有条件,则其可能不会被关注,即要求分支流指明引用语句或守护条件,同样要求引用的语句存在。

规则 8 分支流指明了引用语句或守护条件。

SpecificAlternative. rfs ∪ BoundedAlternative. rfs ∪ GlobalAlternative. guardCondition IsNotNull

规则 9 分支流的引用语句存在。

SpecificAlternative. rfsSentence ∪ BoundedAlternative. rfsSentence IsSubsetOf ConditionCheckSentence

3.6 不一致错误

3.6.1 对象的要求或约束不一致

用例规约对参与者的要求是指特定参与者参与用例功能。用例规约会在两个地方体现这种参与要求:1)事件流中出现该参与者相关动作;2)参与者描述域中存在该参与者。因此,当事件流中出现的参与者未在参与者描述域中出现或参与者描述域中的参与者未参与事件流时,认为存在不一致问题。对应的验证规则为规则 2 与规则 6,即规则 2 与规则 6 所代表的错误是不完整错误,也是不一致错误。

用例规约对其他用例的要求是指特定用例与该用例有包含关系或扩展关系。用例规约会在两个地方体现这种关联要求:1)事件流中的包含用例语句(IncludeSentece)与扩展用例语句(ExtendSentence);2)用例依赖描述域(Dependency)中定义的包含与扩展用例。因此,用例与扩展语句中的用例未在用例依赖描述域中对应出现时认为存在不一致问题。

规则 10 事件流中包含用例语句中的用例在用例依赖的包含用例中存在。

IncludeSentence. usecase IsSubsetOf Dependency. includ

规则 11 事件流中扩展用例语句中的用例在用例依赖的扩展用例中存在。

ExtendSentence. usecase IsSubsetOf Dependency. extend

3.6.2 动作之间在描述逻辑上不一致

用例规约中的每个动作对应事件流中的一个语句。分支流应该引用一个能够引出分支的语句,否则认为分支流要求的引出分支流动作类型与所描述的语句类型冲突。

规则 12 分支流引用语句的类型正确。

SpecificAlternative. rfsSentence ∪ BoundedAlternative. rfsSentence IsSubsetOf ConditionCheckSentence

每个条件检测语句(ConditionCheckSentence)中的条件只有真或假。因此每个条件检测语句只能由一个分支流来处理异常,意味着分支流引用的语句不能重复,否则意味着处理该语句对应的异常存在逻辑冲突。

规则 13 一个条件检测语句不能在多个分支流引用域中被引用。

SpecificAlternative. rfsSentence ∪ BoundedAlternative. rfs Sentence IsUnique

3.6.3 对一个对象的描述与术语的使用不一致

在 RUCM 安全性扩展方法中,试图通过引入数据字典来减少术语使用不一致的问题,并且提供了相应的验证规则。

3.7 形式化规则的验证过程

基于规则的验证过程有 3 个步骤:解析规则,找到规则类别、验证对象和参考对象;找到规则的验证对象对应的需求实例集合,对于特殊的参考对象 Format,直接根据格式名称从 Format 库中取出;执行验证。判断验证对象集合中的每个对象是否满足规则约束,若不满足,则提示出错。

4 案例分析

为了说明方法的有效性,对真实系统的需求进行验证。案例分析的目标有两个:1)说明规则的有效性,通过验证规则能够找到需求中的错误;2)说明规则形式化及验证过程设计的有效性,通过执行依照验证过程逻辑及对应算法编写的程序,输入1)中找到的错误,分析检测出的错误数量说明。

这里选取的案例为3个对需求描述有较高要求的嵌入式系统需求,编写需求的人员包括进行系统开发的相关人员和对RUCM方法进行了学习的研究生。3个案例相互补充,覆盖了验证规则所涉及各类需求项。

高安全机载操作系统需求由经过RUCM方法培训的操作系统需求开发人员按照RUCM方法的模板和约束规则,以文档的形式进行描述,需求人员删减了一些模板内容,比如参与者、用例依赖定义。该系统由于处理的异常情况较多,能很好地体现设计规则中跟分支流相关的规则的验证效果。系统包括86个用例,这里对操作系统的分区管理、空间管理和线程管理3个部分的11个用例进行验证。自动驾驶仪系统需求为真实的某飞机实时操作系统上的一个实时子系统的需求,主要功能是按特定控制率自动控制飞行器的飞行轨迹。这里对其中3个用例进行验证:系统的“主任务”、主任务包含的“数据采集”和“与冗余计算机同步”用例。该案例的特点是用例之间的关系较简单,外部参与者的类别较少。单电梯系统由一个垂直升降梯和多个楼层上的相应设备构成。该案例为课程实践案例,由学习了RUCM建模方法的研究生采用RUCM建模工具编写,该系统的用例主要描述了系统对管理员、乘客的请求处理,有7个用例,其中5个描述了用例规约。该案例的用例事件流结构比较简单,分支流数量较少,用例之间不存在包含或扩展关系,因此对用例依赖和分支流验证的效果不明显,但是外部参与者的数量、种类较多,对于与参与者相关的规则的验证效果较明显。由于在3个系统进行需求建模时尚未对需求描述域的语法格式做严格的限制,因此不针对描述规范符合性的规则进行评估。

通过案例需求审查,发现了44个错误,结果如表2所列。

表2 审查错误个数统计

错误类别	不完整错误个数		不一致错误个数	
	输入错误 个数	检测出的 错误个数	输入错误 个数	检测出的 错误个数
高安全机载操作系统	27	27	3	3
自动驾驶仪系统	2	2	1	1
单电梯系统需求	1	1	10	7
总计	30	30	14	11

经过分析,按照规则能够检查出不同描述特点的需求中的错误,因此设计的验证规则是有效的。统计中出现了异常数据:高安全机载操作系统的不完整错误个数为27,异常原因是需求建模人员在异常处理分支中只输出了错误码,并未指明退出或返回,不出现返回事件的情况,因此都没有以ABORT/RESUME STEP n结束。

基于形式化规则的验证主要是在输入前一节发现的错误的需求项后,统计检测出的错误个数,结果如表3所列。

表3 验证错误个数统计

错误类别	不完整错误		不一致错误	
	输入错误 个数	检测出的 错误个数	输入错误 个数	检测出的 错误个数
高安全机载操作系统	27	27	3	3
自动驾驶仪系统	2	2	1	1
单电梯系统需求	1	1	10	7
总计	30	30	14	11

通过规则验证发现,不一致错误中有3个错误不能被发现,原因有两个:1)术语使用不一致;2)参与者动作出现在全局事件流的守卫条件中。可以看出,验证规则的验证依赖于需求人员对需求的描述方式,这更突出了对需求描述规范性和用词一致性的要求。针对问题1),将通过引入数据字典,验证数据使用的正确性来进行避免。针对问题2),会在以后的研究中进行规则的改进和完善。

通过数据发现,其能够检测出所有的30个不完整问题和79%的不一致问题,因此基于规则的自动化验证方法是可以发现需求中的错误的。

5 相关研究

自动化验证方法中研究比较成熟的是模型验证方法,目前已有许多通用的自动化验证工具,例如JPF,SLAM,SPIN,SMV。几乎所有的自动模型检查工具都有对应要求的验证建模语言,如SPIN要求的验证建模语言是PROMELA,SMV要求输入的是用kripke结构表示的状态图和用CTL逻辑表示的要验证的属性。这些描述语言虽然有效,但是由于形式化程度很高,使得其难以理解;同时对建模人员要求很高,不适用于通常采用自然语言描述的用例需求。而本文选择的RUCM方法通过RUCM建模工具能够获得需求的形式化表达,从而支持需求的自动化验证,既能够保持自然语言的灵活性,同时也能针对RUCM的形式化表达程度进行验证。基于RUCM的形式化表达也有了一些成功的研究^[18,21]。

针对自然语言描述的用例规约,也有一些方法研究了其自动化验证。有些研究^[8,22-23]要求在验证时为用例添加新的信息或依赖于其他模型的信息进行验证。例如文献^[23]提出一种基于标注的用例描述方法,包括流程标注和时间标注,从而支持将需求转成标号迁移系统后交给自动化模型检测工具NuSMV来进行验证。这些方法一方面基于特定的用例规约描述方法,说明针对不同的描述方法有其特定的验证方法;另一方面,从验证模型的角度来确定需求中的错误,例如依据状态机来发现由状态冲突或缺失带来的动作缺失。有的研究基于用例规约的度量值来进行验证,例如文献^[24]通过评估用例中的动作数、参与者动作数、系统动作数来找到需求中潜在的错误,在找出需求的错误后还需要用户进行判断。本文针对特定需求描述方法RUCM,从用例规约的缺陷类型来分析RUCM的错误,从而提出验证规则。

一般的验证方法主要关注的是动作之间的冲突,比如文献^[8]通过在定义用例模型向状态模型的转换,在转换过程中收集状态数据,发现动作之间的冲突和动作缺失;在相同的场景和相同的参与者动作下有不同的系统动作,用例的后置条件不能达。文献^[22]通过结合类图,提出了一个验证审查单,从完整性、正确性和一致性3个方面对用例图和类图进行验

证,用例方面的验证主要针对动作之间的冲突:不同的交互流是否有相同的根操作、有分支的不同守护条件是否会导致分歧等。本文提出的方法不借助其他模型,不仅关注事件流,也关注用例规约中的所有描述域,对一致性、完整性、描述方式3方面的错误类型进行分析,提出验证规则。

文献[4]基于规则来对场景进行验证,找到场景中缺失事件、事件冗余和事件顺序错误3类错误。该方法既可以从即定的规则库中选择规则进行验证,也可以由用户定义规则来进行验证,由此可见基于规则验证的灵活性。本文提出的验证规则也能在实际验证中依据验证需要对规则进行选择。RUCM方法具有良好的扩展性,基于元模型的规则设计以及需求错误的归类,其能够很好地适应RUCM的扩展,以便根据描述方法进行验证规则扩展。

结束语 本文提出了一种基于规则的用例规约验证方法。围绕需求的一致性和完整性,通过分析需求,将用例规约中的错误归类,并基于RUCM用例建模方法来设计验证规则;为了支持自动化验证,形式化地表达了规则,并简单介绍了形式化规则的验证方法。基于规则的设计使得验证方法更灵活,且可以适应RUCM方法的扩展设计。在以后的研究中,应针对扩展的RUCM方法(如针对实时性的RUCM方法)进行规则的扩展设计与验证。

参考文献

- [1] FANMUY G, FRAGA A, LLORENS J. Requirements Verification in the Industry[M]//Complex Systems Design & Management. 2012;145-160.
- [2] YAN Y Q, LI S X, MEI X Y. Analysis of Defect Requirement and Management Model[J]. Computer Science, 2009, 36(4): 140-141. (in Chinese)
严玉清,李师贤,梅晓勇.缺陷需求分析与模型[J].计算机科学,2009,36(4):140-141.
- [3] ARORA C, SABETZADEH M, BRIAND L C, et al. Requirement boilerplates: Transition from manually-enforced to automatically-verifiable natural language patterns[C]//2014 IEEE 4th International Workshop on Requirements Patterns (RePa). IEEE Computer Society, 2014;1-8.
- [4] TOYAMA T, OHNISHI A. Rule-based Verification of Scenarios with Pre-conditions and Post-conditions[C]//2013 21st IEEE International Requirements Engineering Conference (RE). IEEE Computer Society, 2005;319-328.
- [5] JOHNSON L A. Software Considerations in Airborne Systems and Equipment Certification; Document RTCA-DO-178B[R]. 1992.
- [6] KOESTERS G. Coupling Use cases and Class Models as a Means for validation and verification of requirements Specification[J]. Requirements Engineering, 2001, 6(1):3-17.
- [7] LEITE J C S D P, HADAD G D S, DOOM J H, et al. Scenario Construction Process [J]. Requirements Engineering, 2000, 5(1):38-61.
- [8] SOMEÉ S S. Supporting use case based requirements engineering[J]. Information & Software Technology, 2006, 48(1):43-58.
- [9] TERNER F, IVARSSON M, PETTERSSON F, et al. Defects in automotive use cases[C]//Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering. ACM, 2006;115-123.
- [10] JIN N, YANG J. An Approach of Inconsistency Verification of Use Case in XML and the Model of Verification Tool[C]//2010 International Conference on Multimedia Information Networking and Security. IEEE Computer Society, 2010;757-761.
- [11] IEEE I. Recommended practice for software requirements specifications; IEEE Std 830-1998[S]. 1998;1-40.
- [12] KAMSTIES E, PAECH B. Taming ambiguity in natural language requirements[C]//ICSSEA 2000. 2000;1-8.
- [13] TJONG S F, HALLAM N, HARTLEY M. Improving the quality of natural language requirements specifications through natural language requirements patterns [C] // IEEE International Conference on Computer and Information Technology. 2006.
- [14] FABBRINI F, FUSANI M, GNESE S, et al. The Linguistic Approach To The Natural Language Requirements Quality: Benefit of The Use of An Automatic Tool[C]//Proceedings of Software Engineering Workshop, 2001. 2001;97-105.
- [15] JURISTO N, MORENO A M, Lopez M. How to use linguistic instruments for object-oriented analysis [J]. IEEE Software, 2000, 17(3):80-89.
- [16] DURÁN A, BERNÁRDEZ B, TORO M, et al. Expressing Customer Requirements Using Natural Language Requirements Templates and Patterns[C]//IMACS/IEEE CSCC'99. Athens, IMACS/IEEE, 1999.
- [17] DINGER C. High Quality Requirements Specifications for Embedded Systems through Authoring Rules and Language Patterns [J/OL]. <http://publica.fraunhofer.de/documents/N-10469.html>.
- [18] YUE T, BRIAND L C, LABICHE Y. Automatically Deriving a UML Analysis Model from a Use Case Model; Technical Report [R]. Carleton University, 2013.
- [19] ZHANG G, YUE T, ALI S. Modeling Crisis Management System with the Restricted Use Case Modeling Approach[C]//CMA @ MODELS. Miami, FL, USA, 2013;1-8.
- [20] ZHANG G, YUE T, WU J, et al. Zen-RUCM; A Tool for Supporting a Comprehensive and Extensible Use Case Modeling Framework[OL]. <http://www.ceur-ws.org/vol-1115/demo8.pdf>.
- [21] ZHANG M, YUE T, ALI S, et al. A Systematic Approach to Automatically Derive Test Cases from Use Cases Specified in Restricted Natural Languages[M]//System Analysis and Modeling: Models and Reusability. Springer International Publishing, 2014;142-157.
- [22] KESTERS G, SIX H W, WINTER M. Coupling Use Cases and Class Models as a Means for Validation and Verification of Requirements Specifications[J]. Requirements Engineering, 2001, 6(1):3-17.
- [23] SINHA A, SUTTON S M, PARADKAR A. Text2Test: Automated Inspection of Natural Language Use Cases[C]//Third International Conference on Software Testing, Verification and Validation. IEEE, 2010;155-164.
- [24] JIN N, YANG J. An Approach of Inconsistency Verification of Use Case in XML and the Model of Verification Tool[C]//2010 International Conference on Multimedia Information Networking and Security. IEEE Computer Society. 2010;757-761.