

面向问题的软件开发协同建模工具的研究与实现

张 晓¹ 李 智^{1,2} 赵子岩¹ 付昌兰¹ 李伟东¹ 禹月昆¹ 王 超¹

(广西师范大学计算机科学与信息工程学院 广西 桂林 541004)¹

(广西师范大学广西多源信息挖掘与安全重点实验室 广西 桂林 541004)²

摘 要 建模软件是辅助需求工程师分析的工具,在需求设计阶段必不可少。目前,很少有需求建模工具可以跨平台运行,支持在线多用户协同以及验证需求模型正确性和完整性等功能。鉴于问题框架方法在需求工程领域获得了较大关注,文中开发了一款用户体验较好、兼容多平台的计算机辅助问题框架建模软件。本研究解决了自动化校验问题图的正确性和完整性以及复杂问题图拆分等难题,实现了用户登录、云端数据库存储设计和多人协同建模和验证,从而构建了一个在线需求建模、共享和验证的平台。

关键词 建模软件,问题框架,问题图拆分,协同建模

中图分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.09.018

Research and Implementation of Collaborated Modeling Approach for Problem-oriented Software Development

ZHANG Xiao¹ LI Zhi^{1,2} ZHAO Zi-yan¹ FU Chang-lan¹ LI Wei-dong¹ YU Yue-kun¹ WANG Chao¹

(College of Computer Science and Information Technology, Guangxi Normal University, Guilin, Guangxi 541004, China)¹

(Guangxi Key Laboratory of Multi-source Information Mining & Security, Guangxi Normal University, Guilin, Guangxi 541004, China)²

Abstract Software modeling tools are essential for assisting requirements engineers in system analysis during the requirements and design phase. At present, few existing requirements modeling tools can be run across different platforms, support online multi-user collaborations, and verify the correctness and completeness of requirements models. As the problem frames(PF) approach attracts much attention in the requirements engineering community, a computer-aided PF modeling tool, which provides good user experience and is compatible with multiple platforms, was developed in this paper. This work solves two difficult problems, i. e., automatic verification of the correctness and completeness of problem diagrams and mechanized decomposition of complex problem diagrams. Therefore, an online platform for requirements modeling, sharing and verification was established to support multi-user logins, deploy databases in the clouds and facilitate multi-user collaborations.

Keywords Modeling software, Problem frames, Problem diagram decomposition, Collaborated modeling

1 引言

需求分析作为软件开发过程中至关重要的一环,正日益受到人们的重视和关注。根据研究发现,软件项目中 40%~60% 的错误都是在需求分析和获取阶段产生的^[1]。美国权威组织 Standish Group 曾经对重大 IT 项目进行了调查研究,其中 2001 年的报告显示,有多达 70% 的软件项目是以失败告终的,而导致这些项目失败的最主要原因是需求分析阶段的规格说明不完整。因此,需求分析的重要性不言而喻,软件建

模方法由此也得到了广泛的重视,更有公司为统一建模语言(UML)和面向目标的 KAOS 方法开发了相应的辅助支持工具,提供给开发人员使用。

很多软件密集型系统的失效是由对其组件边界不正确的假设所导致的。随着部署的软件规模的增加,系统变得越来越复杂和精细,从而增加了确定组件边界假设的难度。这种情况下,即使所有的组件都满足各自的规约,但如果这些规约结合起来不能满足利益相关者提出的端到端需求,那么所开发的系统的正确性将无法保证^[2]。

到稿日期:2017-10-15 返修日期:2017-12-06 本文受广西研究生教育创新计划项目(XYCSZ2017066),学位与研究生教育改革课题(XJGY201809),校级双语教学课程建设项目(A-0201-00-00013F),国家自然科学基金项目(61262004),广西区域多源信息集成与智能处理协同创新中心,广西多源信息挖掘与安全重点实验室 2014 年度主任基金课题(14-A-03-01),2015 年系统性研究课题(15-A-03-01),广西科学研究与技术开发计划项目(桂科合 1347004-22),广西自然科学基金项目(2012GXNSFCA053010),广西科技基地和人才专项(桂科 AD16380008)资助。

张 晓(1991-),男,硕士,主要研究方向为软件开发方法与理论,E-mail:494471788@qq.com;**李 智**(1969-),男,博士,教授,CCF 高级会员,主要研究方向为软件需求工程、经验软件工程和软件测试,E-mail:zhili@gxnu.edu.cn(通信作者);**赵子岩**(1991-),男,硕士,主要研究方向为软件需求工程;**付昌兰**(1990-),女,硕士,主要研究方向为软件需求工程;**李伟东**(1989-),男,硕士,主要研究方向为软件需求工程;**禹月昆**(1991-),女,硕士,主要研究方向为软件需求工程;**王 超**(1993-),男,硕士,主要研究方向为软件需求工程。

问题框架^[3]提供了一种方法来描述软件与其他系统之间的交互。它帮助开发者理解软件问题所在的上下文,且其在很多方面与解决方案的设计相关。问题框架强调将端到端的软件需求分解为机器规约和一组领域属性。该方法由软件工程领域著名学者 Jackson 最先提出^[3],后来经过不断的研究和发展,问题框架建模语言更加完善。相对于其他软件开发方法,问题框架方法充分考虑了周围环境对软件系统的影响,对现实世界进行刻画,并且把需求的指称含义落实到现实世界相关领域的描述上,因此越来越多的研究者开始关注和研究问题框架,并取得了一定的进展。但计算机辅助支持工具仍有欠缺,特别在协同建模和模型检查方面缺少实用工具的支持,因此完善和扩展问题框架辅助工具集,对于评价该方法有着研究和应用价值。

传统的建模工具在协同和自动化检验图的完整性和正确性上存在不足,例如:MS Visio, Rational Rose 和 UMLet 没有实现多用户之间共享文件和协同建模的功能,对绘制出来的需求设计图无法判断完整性和正确性。本文将重点介绍基于问题框架建模的计算机辅助支持工具及其在建模软件中的实现。

2 完整性、正确性检查研究

问题框架方法^[4]建模的模型定义了机器领域(machine domain,用□表示)、问题领域(problem domain,用□表示)和需求(requirement,用○表示)。其中问题领域又进一步分为设计领域(design domain,用□表示)和给定领域(given domain)。

机器领域与问题领域之间的连线称为接口交互(interface,用—表示),需求与问题领域之间的连线称为需求引用(reference,用……表示),需求与问题领域之间的连线称为需求约束(constraint,用-->表示)。

2.1 OCL 简述

对象约束语言(Object Constraint Language, OCL),是一种在指定的模型元素上施加约束的语言。OCL 表达式以附加在模型元素上的条件和限制来表现对该对象的约束,其中包括附加在模型元素上的不变量或约束的表达式、附加在操作和方法上的前置条件和后置条件等^[6-7]。

1997年,UML采用对象约束语言(OCL)来定义约束。问题框架同UML一样,也可以使用OCL来定义。接下来本文对问题图的完整性和正确性约束进行叙述。

2.2 问题图的完整性、正确性条件

问题图的不完整性是指在某些设计图中表现出来的信息不完善、缺少模型元素,领域节点和需求节点中缺少接口、约束和引用等,造成绘制的图形不符合最基本的问题框架图规范。问题图的完整性条件如表1所列。

表1 问题图的完整性条件

Table 1 Completeness conditions for problem diagram

编号	完整性条件
1	领域的名称必须被设置
2	上下文图中至少有一个机器领域
3	一个问题图/框架至少有一个需求
4	领域的缩写必须被设置
...	...

问题图的正确性是指在设计图中模型元素之间需要满足问题框架方法的约束。问题图的正确性条件如表2所列。

表2 问题图的正确性条件

Table 2 Correctness conditions for problem diagram

编号	正确性条件
1	领域的名称必须唯一
2	需求不能直接约束机器领域
3	每个机器领域至少控制一个接口
4	领域的缩写必须唯一
...	...

当人们对需求进行建模时,希望软件能够帮助需求分析人员检查设计是否完整和正确,这在一定程度上避免了由设计不规范而导致的开发人员难以理解设计的情况的出现。下面举例说明不完整和不正确的问题图,如图1和图2所示。图1中的问题领域没有名称,图2中的需求领域和机器领域之间不能存在约束。

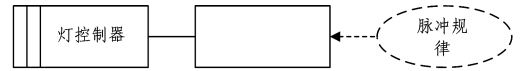


图1 交通灯问题图(领域无名称)

Fig. 1 Traffic lights: problem diagram (domain without a name)

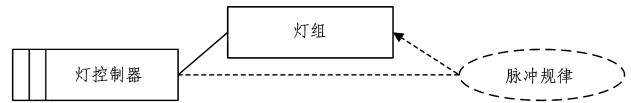


图2 交通灯问题图(需求直接约束机器领域)

Fig. 2 Traffic lights: problem diagram (requirement directly constraining machine domain)

本文介绍的建模软件实现了问题图的完整性和正确性检查,给需求分析人员带来了极大的方便。本文研究借鉴 OCL 在 UML 模型上的一致性验证方法,用 OCL 定义问题框架约束,并通过分析 OCL 表达式来开发模型的完整性、正确性检查功能模块^[5]。同时,因为实现验证的代码太长且不易理解,本文将使用 OCL 来描述问题框架模型约束。

2.3 OCL 描述完整性、正确性检查功能

首先,声明 OCL 表达式中的变量 Class 和 Interface,其分别表示包含所有的领域和接口的类。限于篇幅,使用 OCL 来描述表1和表2中的问题图约束功能。

约束条件:领域的名称必须被设置

OCL 表达式:

Class.allInstances() -> select (c | c.oclAsType (Class).getValue (c.oclAsType (Class).getAppliedStereotypes () -> asSequence () -> first (), 'value') = null) -> size () = 0

约束条件:上下文图中至少有一个机器领域

OCL 表达式:

Class.allInstances () -> forAll (p | p.oclAsType (Class).getAppliedStereotypes ().name -> includes ('Machine') -> size () >= 1)

约束条件:一个问题图/框架至少有一个需求

OCL 表达式:

Class.allInstances () -> forAll (p | p.oclAsType (Class).getAppliedStereotypes ().name -> includes ('Requirement') -> size () >= 1)

约束条件:领域的缩写必须被设置

OCL 表达式:

```
Class, allInstances() -> select(c | c.oclAsType(Class),
getValue(c.oclAsType(Class).getAppliedStereotypes() ->
asSequence() -> first(), 'abbreviation') = null) -> size() = 0
```

约束条件:领域的名称必须唯一

OCL 表达式:

```
Class, allInstances() -> select(getAppliedStereotypes(),
name -> includes('Domain')) -> isUnique(name)
```

约束条件:需求不能直接约束机器领域

OCL 表达式:

```
Interface, allInstances() -> select(a | a.oclAsType(Dependency),
getAppliedStereotypes(), name -> includes('constrains')) ->
forall(source, getAppliedStereotypes(), name ->
includes('Requirement')) implies not target, getAppliedStereotypes(),
name -> includes('Machine'))
```

约束条件:每个机器领域至少控制一个接口

OCL 表达式:

```
Interface, allInstances() -> select(getAppliedStereotypes(),
name -> includes('observes')), target -> forall(ot |
Interface, allInstances() -> select(getAppliedStereotypes(),
name -> includes('controls')) -> select(target -> exists
(ct | ct = ot)) -> size() = 1)
```

约束条件:领域的缩写必须唯一

OCL 表达式:

```
Class, allInstances() -> select(c | c.oclAsType(Class),
getValue(c.oclAsType(Class).getAppliedStereotypes() ->
asSequence() -> first(), 'abbreviation')) -> isUnique
(oclAsType(String))
```

以上是第 2.2 节中提到的部分条件对应的 OCL 表达式,本文的软件实现了检查问题图是否满足这些约束条件的功能。

3 复杂问题拆分研究及实现

Hall^[8]为问题框架提供的一个指称语义可以用来表示整个问题图。问题图可以表示为:

$$c, o: [K, R] = \{S | S \text{ controls } c \wedge S \text{ observes } o \wedge K, S \vdash_{DRDL} R\}$$

图 3 是对应的通用问题图(注: $S!c$ 表示“S controls c”, $S?o$ 表示“S observes o”):

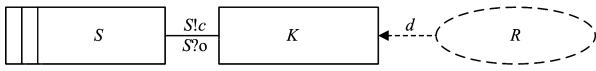


图 3 通用问题图

Fig. 3 Generic problem diagram

本文引入因果关系链,以便对复杂问题图进行理解,在软件中实现因果关系的自动化查找^[9]。

3.1 因果关系链

由图 3 可以看出,如果领域共享现象 o 导致了 c 的发生,那么这是问题图中的因果关系。

在一个问题图中,如果找到一条 $R \rightarrow S$ 或 $R \rightarrow S \rightarrow R$ 且含有 o 到 c 的路径,则说明找到了问题的一个解,这条路径上的元素构成了问题的解的集合。一个复杂问题图中会有多条这样的路径,也即有多条因果关系链。其中,每条路径中的 R, S 可以分别表示为 $R_1, S_1, R_2, S_2, \dots, R_n, S_n$ 。

引入因果关系链概念后的 Hall 指称语义表达式如下:

$$c, o: [K, R] = \{S | S!c \wedge S?o \wedge K, S \vdash_{DRDL} R\} \\ = \{S_1 \parallel S_2 \parallel \dots \parallel S_n\}$$

图 4 对应的是一个部分问题图,图中显示的是一条解的集合。

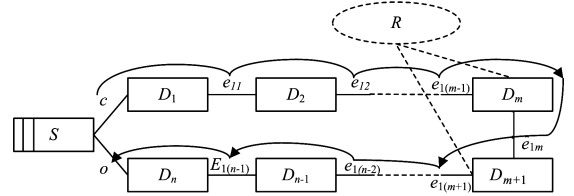


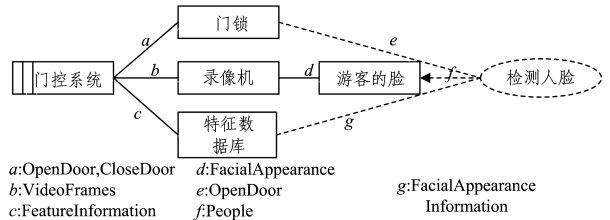
图 4 部分问题图

Fig. 4 Partial problem diagram

3.2 安全门控制问题图案例分析

本文使用安全门控制问题图案例分析,使读者更加理解因果关系链和 Hall 的指称语义表达式。

需要一台识别人脸特征的计算机来控制安全门,想要进入安全门的每个人的脸都被捕获到一个录像带上,使用数据库中的项与捕获的人脸特征进行比较,数据库中的项是已经被明确可以进入的人脸特征,图 5 是安全门控制问题图。



a:OpenDoor,CloseDoor d:FacialAppearance
b:VideoFrames e:OpenDoor
c:FeatureInformation f:People g:FacialAppearance Information

图 5 安全门控制问题图

Fig. 5 Secure door control problem diagram

通过比较图 3 和图 5 可以发现问题图与 Hall 指称语义的关系以及 2 条因果关系链,如图 6 所示。

$S = \{\text{门控系统}\}; c = \{a\}$ (b 被 S 初始化,因此受 S 的控制); $o = \{b, c\}$ (b, c 被 S 接收,因此受 S 的监视); $K = \{\text{门锁, 录像机, 游客的脸, 特征数据库}\}; R = \{\text{检测人脸}\}; d = \{e, f, g\}$ 。

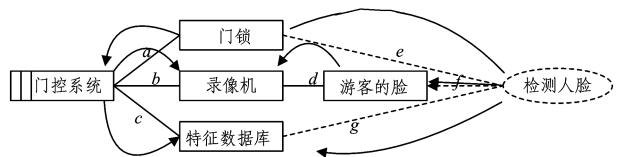


图 6 安全门控制问题图因果关系链

Fig. 6 Causal chains in secure door control problem diagram

4 多平台的建模软件的设计及功能

本文中的问题框架建模软件采用的是客户端/服务器(Client/Server, C/S)和浏览器/服务器(Browser/Server, B/S)

混合体系架构。因为两种体系架构各有优点,所以混合体系架构能够节省开发和维护成本,使系统具有良好的开放性,易于扩展和移植等^[10]。C/S和B/S混合的软件体系结构如图7所示。同时,该建模软件集成了用户登录模块,用户在离线或在线状态时都可以使用软件,这使得本建模软件在有无网络下都可以使用。有网络的情况下,用户的数据可以永久保存在云端服务器数据库中;同时可实现多人协同建模功能,方便多个需求分析师不受地域限制和客户端软件的限制共同建模,从而带来更好的体验。

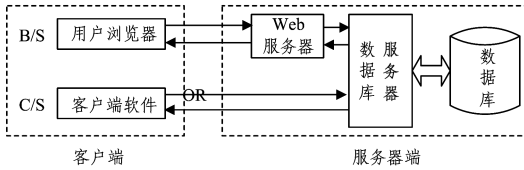


图7 问题框架协同建模的软件体系结构图

Fig.7 Software architecture for PF collaborated modeling

4.1 软件协同建模功能的实现

由于多人协同建模,因此难免会涉及设计上的分歧和同一个设计被其他用户篡改的冲突问题。因此本软件设计了用户模块,每个用户都必须在软件官网注册帐号才可以使用在线建模功能,没有帐号的用户仅可以在本地使用,不能共享和保存设计。一个用户可以将自己的设计共享给其他人员并使其参与设计,其他人员的设计也会在数据库中同步。同时软件也会保留旧的设计,从而达到版本控制的目的。图8—图10是界面截图。



图8 客户端软件用户登录界面

Fig.8 User login interface at client side



图9 浏览器访问用户登录界面

Fig.9 User login interface using browser

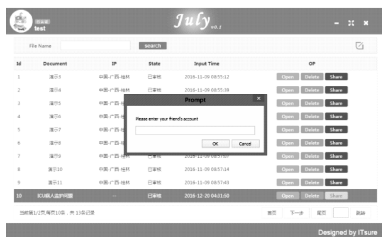


图10 共享设计图给其他用户

Fig.10 Sharing design diagrams with other users

4.2 软件的主要功能

软件除了具有基本的绘制问题图的功能以外,还实现了完整性、正确性校验问题图的功能,也能够根据因果关系找到因果关系链。绘制好的问题图是XML格式的,用户可以上传自己的工程图到云服务器数据库中,需要修改时,再从数据库中打开。本文首先使用软件绘制安全门控制问题图(见图11),使软件自动校验图的正确性和完整性。

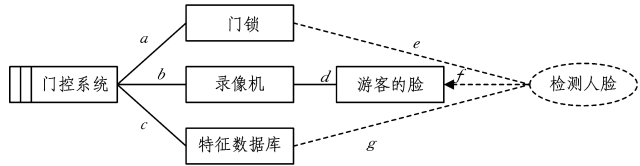


图11 绘制正确的问题图

Fig.11 Correct problem diagram

绘制好问题图,点击软件菜单中的OCL字样的按钮,即可以自动化校验问题图是否完整和正确(见图12)。

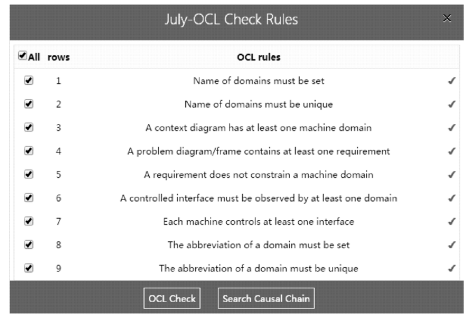


图12 校验问题图结果截图

Fig.12 Snapshot of verification results of problem diagram

如果绘制不完整的或不正确的问题图(见图13),那么图中问题图的一个领域缺少名称。将绘制好的问题图进行完整性校验,结果发现了这个错误,如图14所示。

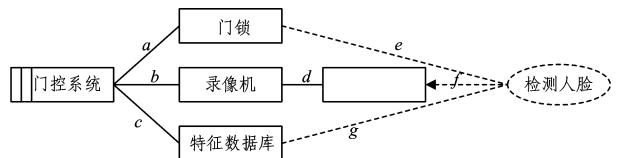


图13 绘制错误的问题图

Fig.13 Incorrect problem diagram

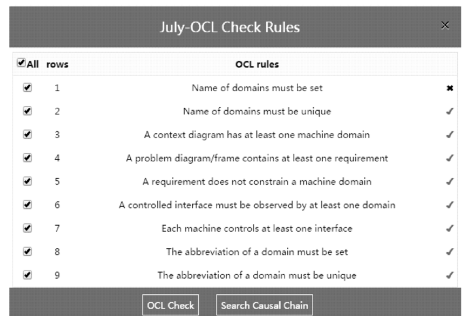


图14 校验问题图结果截图(符号“×”代表发现一个错误)

Fig.14 Snapshot of verification results of a problem diagram

(“×” symbol represents error is found)

Cognitive Radio Systems[D]. Beijing: Beijing University of Posts and Telecommunications, 2011. (in Chinese)

陈斌华. 认知无线电系统中的频谱预测算法研究[D]. 北京: 北京邮电大学, 2011.

- [14] WU J R, HU J M, QIN J X. Recognition Radio Spectrum Prediction Based on K-RBF Neural Network[J]. Technology of Television, 2014, 38(5): 105-108. (in Chinese)

吴建斌, 胡津铭, 秦继新. 基于 K-RBF 神经网络的认知无线电频谱预测[J]. 电视技术, 2014, 38(5): 105-108.

- [15] ZHANG K, QI L N. Collaborative Spectrum Detection Based on Continuous Hidden Markov Model[J]. Computer Technology & Development, 2015, 25(6): 64-68. (in Chinese)

张凯, 齐丽娜. 基于连续隐马尔科夫模型的协作频谱检测[J]. 计算机技术与发展, 2015, 25(6): 64-68.

- [16] LING X, WU B, WEN H, et al. Adaptive Threshold Control for

Energy Detection Based Spectrum Sensing in Cognitive Radios [J]. IEEE Wireless Communications Letters, 2012, 1(5): 448-451.

- [17] DIGHAM F F, ALOUINI M S, SIMON M K. On the Energy Detection of Unknown Signals Over Fading Channels[J]. IEEE Transactions on Communications, 2007, 55(1): 21-24.

- [18] XING X S. Research on Spectrum Prediction in Cognitive Radio Network [D]. Beijing: Beijing Jiao Tong University, 2014. (in Chinese)

邢晓双. 认知无线网络中的频谱预测技术研究[D]. 北京: 北京交通大学, 2014.

- [19] NI S, BAI X, WANG Z, et al. A new method of cognitive radio spectrum prediction research [C] // International Congress on Image and Signal Processing, Biomedical Engineering and Informatics. IEEE, 2017: 982-986.

(上接第 122 页)

经软件检验后的正确问题图, 可再用软件从问题图中找出所有的因果链条(见图 15)。

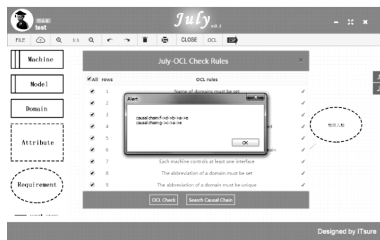


图 15 查找因果链条结果截图

Fig. 15 Snapshot of searching and finding causal chains

在安全门控制案例中, 共享现象 o 的集合中有 b 和 c 两个共享现象。共享现象 c 的集合有一个共享现象 a 。从而可以找到 2 条从需求到机器领域再到需求且包含 o 到 c 的路径, 即因果链条, $f \rightarrow d \rightarrow b \rightarrow a \rightarrow e$ 和 $g \rightarrow c \rightarrow a \rightarrow e$ 。

由软件给出的结果可知, 安全门控制需求模型中有 2 条因果链条, 与第 3 节案例中分析的结果一致。

结束语 本文提供了自动化校检问题图和查找复杂问题的子问题解, 即因果链条。通过研究传统建模工具的不足, 为提高需求设计的准确性, 尽可能地规避因需求阶段产生的错误而导致软件项目失败的情况, 本文设计的软件可以实现多人协同完成同一个项目的需求建模。问题框架方法比较适合应用于信息物理融合系统^[11], 而对于信息物理融合系统的设计是否建立在端到端的需求上的问题, 需要我们引用因果链条的成果, 将问题图中的因果链条转化成形式化脚本语言, 进行需求满足性校验来解决。因此, 该软件可以更好地帮助需求工程师理解复杂问题和验证设计。

参考文献

- [1] CLELAND-HUANG J, CHANG C K, CHRISTENSEN M. Event-Based Traceability for Managing Evolutionary Change [J]. IEEE Transactions on Software Engineering, 2003, 29(9): 796-810.

- [2] SEATER R, JACKSON D. Problem frame transformations: deriving specifications from requirements [C] // International Workshop on Advances and Applications of Problem Frames. ACM, 2006: 71-80.

- [3] JACKSON M. 软件开发问题框架: 现实世界问题的结构化分析: analyzing and structuring software development problems [M]. China Machine Press, 2005.

- [4] CHEN X H, YIN B, JIN Z. Ontology-Guided Requirements Modeling Based on Problem Frames Approach [J]. Journal of Software, 2011, 22(2): 177-194. (in Chinese)

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 [J]. 软件学报, 2011, 22(2): 177-194.

- [5] QUERALT A, TENIENTE E. Verification and Validation of UML Conceptual Schemas with OCL Constraints [J]. Acm Transactions on Software Engineering & Methodology, 2012, 21(2): 1-41.

- [6] RICHTERS M, GOGOLLA M. On formalizing the UML object constraint language OCL [C] // International Conference on Conceptual Modeling. Springer, Berlin, Heidelberg, 1998: 449-464.

- [7] JIANG Z F, WANG L Z, LI X D, et al. An Approach to Generate Test Cases Based on UML Sequence Diagrams [J]. Computer Science, 2004, 31(7): 131-136. (in Chinese)

江泽凡, 王林章, 李宣东, 等. 基于 UML 顺序图的测试方法 [J]. 计算机科学, 2004, 31(7): 131-136.

- [8] LI Z, HALL J G, RAPANOTTI L. On the systematic transformation of requirements to specifications [J]. Requirements Engineering, 2014, 19(4): 397-419.

- [9] JACKSON M. Where, Exactly, Is Software Development? [M] // Formal Methods at the Crossroads. From Panacea to Foundational Support. Springer Berlin Heidelberg, 2003: 115-131.

- [10] CHEN X, LIU J L. Analysis and Comparison between the Structures of Client/Server and Browser/Server [J]. Journal of Chongqing Institute of Technology Management, 2000, 14(2): 100-103.

- [11] JACKSON M. System Behaviours and Problem Frames: Concepts, Concerns and the Role of Formalisms in the Development of Cyber-physical Systems [M]. Dependable Software Systems Engineering, 2015: 79-104.