

DOI: DOI:10.3969/j.issn.1673-4785.201409015

网络出版地址: http://www.cnki.net/kcms/doi/10.3969/j.issn.1673-4785.html

扩展断言知识检验一致的需求建模方法

郭新峰¹, 马世龙², 吕江花², 李睿²

(1. 山西大学 软件学院, 山西 太原 030013; 2. 北京航空航天大学 计算机学院, 北京 100191)

摘要: 保证需求完整一致是需求建模的关键, 文中提出一种基于知识库推理检验一致性的需求建模方法, 该方法在建模中引入本体以增强表达能力, 并结合语义建立约束规则构建需求模型知识库, 基于描述逻辑描述需求和检验需求一致性, 将需求建模过程作为断言知识库的扩展过程, 把需求完整性问题转化为需求一致性问题, 并归结其为规则约束下的可满足问题。利用 Tableau 算法等实现需求一致性的检验, 在整个建模活动中保证需求一致。应用实践中使用自行编制的 ROB 半自动建模软件进行需求建模与分析, 验证了建模方法的需求一致性, 并通过案例说明了扩展断言知识库检验一致的需求建模过程。

关键词: 需求模型; 一致性; 完整性; 多视点; 描述逻辑; 知识库; 可满足性; Tableau; 需求工程

中图分类号: TP311 **文献标志码:** A **文章编号:** 1673-4785(2015)01-0081-10

中文引用格式: 郭新峰, 马世龙, 吕江花, 等. 扩展断言知识检验一致的需求建模方法[J]. 智能系统学报, 2015, 10(1): 81-89.

英文引用格式: GUO Xinfeng, MA Shilong, LYU Jianghua, et al. Requirement modeling method for inspecting the consistency of the expanded asserted knowledge [J]. CAAI Transactions on Intelligent Systems, 2015, 10(1): 81-89.

Extension abox requirements modeling method

GUO Xinfeng¹, MA Shilong², LYU Jianghua², LI Rui²

(1. College of Software, Shanxi University, Taiyuan 030013, China; 2. Computer College, Beihang University, Beijing 100191, China)

Abstract: Assuring a complete and consistent requirement is the key to requirement modeling. This paper proposes a requirement modeling method for inspecting consistency on the basis of knowledge base reasoning. An ontology is added into the modeling process to improve the expression ability. In addition, the semantic establishment constraint rule is used to build the knowledge base of requirement modeling. The requirement modeling process is taken as an expansion process of the assertion knowledge base based on the consistency among the descriptive logic, descriptive requirement and inspection requirement. The matter of requirement completeness is converted into requirement consistency and attributed to the satisfiability under the constraint by rules. Tableau algorithm is utilized to realize the inspection on requirement consistency, assuring the requirement to be consistent in the entire modeling process. In the application practice, the self-programmed ROB semiautomatic modeling software was applied to requirement modeling and analysis to verify the requirement consistency of the modeling method. In addition, the requirement modeling process on detecting the consistency of an expanded asserted knowledge base was shown by cases.

Keywords: model; consistency; completeness; multi-view; description logic; knowledge base; satisfiability; Tableau; requirement engineering

软件依据需求而建模, 需求建模目的是抽取和验证需求, 以期真实可靠、稳定。需求是人们对真实世界现象的期望, 由于人类认识常存在矛盾, 需求不

一致难以避免: 1) 需求冲突、语义矛盾; 2) 需求信息不完整、模糊; 3) 需求冗余是需求不一致的根本原因^[1-2]。目前相关研究有一些成果, 如 Nuseibeh 等提出不一致需求管理框架^[3], 认为不一致的需求是不满足规则的命题, 而任何需求建模方法隐含一组检验规则; 还有一些相关研究是多视点需求建模的不一致管理^[4], 认为需求的不一致源自不同角度的认识和表达, 从不同视点将系统分解为多个局部视

收稿日期: 2014-09-09 网络出版日期: 2015-01-13

基金项目: 国家自然科学基金资助项目(61300007, 61003016); 软件开发环境国家重点实验室开放基金资助项目(SKLSDE-2012ZX-28, SKLSDE-2013ZX-11)。

通信作者: 郭新峰. E-mail: rjxy_gxf@sxu.edu.cn.

图,分别获取需求完整建模,在集成过程中检测不一致;而王学智等对 C4ISR 系统能力需求中存在的_{不一致问题}深入分析,构建能力模糊本体,研究了能力需求模型一致性验证技术^[5]。

需求不一致的处理方法主要有:1)逻辑推理证明方法采用带标记的逻辑公式作为需求表示形式,通过禁止所有平凡推理暂时容忍不一致需求,然后维护需求一致性断言集并设置需求优先级别,检测、跟踪并自动处理不一致需求;2)模型检验方法采用状态变迁作为需求建模原则,并利用模型检验技术进行需求不一致性处理;3)基于目标的方法区分了需求不一致的4种程度,包括矛盾、分歧、竞争和障碍,通过检测和处理分歧解决不一致问题。这些研究成果已有很多应用,在一些特定领域取得很好效果,为相关研究做出很多有益指导,但这一过程对建模人员有较高的知识要求,缺少更接近普通认知的过渡,进一步的应用受到很大限制,其根本原因是不能将易理解方便沟通与高度自动化形式化2个要求统一起来,导致需求模型难以与实际对应,建模理论难以转换为实际应用,需求分析管理自动化程度和智能化水平不高。基于资源、组织与业务的需求模型(requirement model on resources, organization and business, ROB),用人们认识世界的一般方法抽取需求,将数据、角色和功能指称到描述真实世界的资源、组织与业务,从不同视角基于递归分解思想将需求抽象为需求树,以期确定、完整一致的表达需求。文中在 ROB 基础上提出建立 ROB 知识库,通过知识推理检验需求一致性,并将一致需求增加到断言知识库中,形成逐步扩展断言库的需求建模方法,保证了需求完整性、一致性。

1 ROB 结构

ROB 从 3 个视点(资源、组织与业务)进行需求分析,形成多视点结构:1)业务视点通过分类、分解的方法分析业务以描述功能需求;2)组织视点分析角色权限和人为活动等需求;3)资源视点获取数据需求。

ROB 具有层次结构:1)资源层建立数据模型形成数据需求;2)组织层将参与者抽象为角色,通过权限、关系建立需求模型;3)业务层为真实模拟业务活动过程,从业务分解与协同(角色、资源与业务协作)建立功能需求模型。ROB 层次间具有关联。

ROB 基于递归分解的思想:1)业务分为复合业务和活动;复合业务由一组更基本的业务构成,这些基本业务之间存在线性、并行等关系,即业务流;活动是最基本的业务,包括角色、资源和 API;API 定义为五元组,包括前置条件、起始状态、过程状态、终止状态和目标状态。2)组织分为机构组织和角色;机构组织由更基本的机构组织构成,这些基本组织

间存在辖制关系;角色是最基本的组织,角色在特定资源上具有一组操作能力。3)资源分为复合资源和原子资源;复合资源可由更基本的资源构成,这些基本资源可分为主资源和非主资源,二者有确定的函数依赖关系;原子资源是最基本的资源,由一组属性组成,包括主属性和非主属性,主属性和非主属性间存在确定的函数依赖关系。ROB 将需求抽象为统一的树形模型,如图 1 所示。

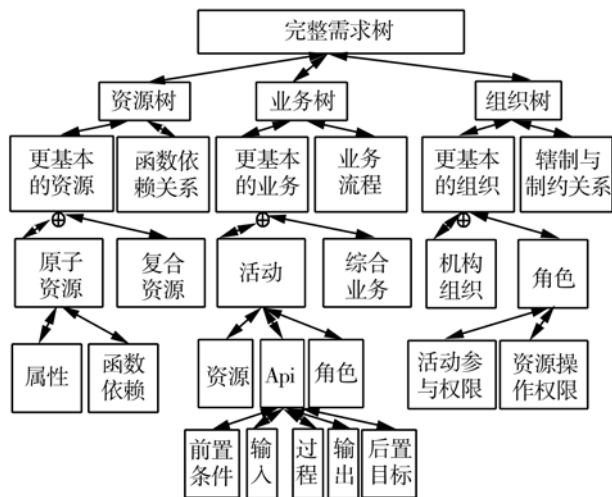


图 1 ROB 模型树形结构

Fig.1 Tree-shaped diagram of ROB-model

2 ROB 本体概念体系

从需求理解的角度看,需求工程依赖于人们对现实世界的认识程度,现实世界被抽象为概念与关系,因此在需求工程中引入本体增强了表达能力,在很大程度上满足了需求分析的要求。本体是相关专题的基本术语和关系,以及利用这些术语和关系构成该专题的规则集合^[6],本体是领域概念模型的显示表示^[7]。陆汝钤等将对象和本体结合,提出面向本体的需求分析 OORA^[8],此后本体论应用于需求工程的研究有了很大发展,如金芝等提出的基于问题框架的需求建模^[9]。

2.1 ROB 本体概念层次

本体概念是 ROB 建模全过程所有方法及对象分类的概括,主要包括需求任务(Errand)、业务功能(Business)、执行模块(Task)、业务状态(State)、组织与角色(Organization)、资源(Resource)、需求任务类型(ERdType)、资源类型(RscType)和操作类型(OptType)。ROB 本体概念层次结构如图 2 所示,T 表示 ROB 本体概念全域。Errand 分为基本简单任务(Basic)和复合任务(Composite);Task 分为前置模块(Pretask)和核心模块(Stage);State 分为前置条件(Precondit)、输入(Input)、过程(Procedure)、输出(Output)和后置服务(Postcondi);Organization 分为部门(Department)和角色(Role),Department 分为领导(Leader)和员工(Subject);Resource 分为元素(El-

ment)和属性(Attribute), Element 分为主成员(Key) 和普通成员(Member); Attribute 有一组操作权限。

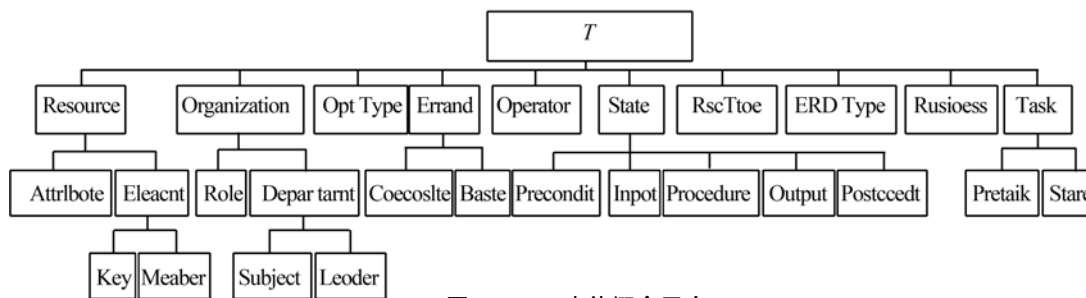


图2 ROB 本体概念层次

Fig.2 Concept categories of ROB ontology

2.2 ROB 模型本体关系

ROB 本体中概念不是孤立存在的,概念间存在关系,主要包括:1) isA,描述格式为 isA (A, B),表示 A 是 B 的子类 (subclass); 2) hasRsc, 描述格式为 hasRsc (A, B),语义解释为一个 A (需求任务/角色/状态)涉及到一个资源 B ; 3) hasOrg, 描述格式为 hasOrg (A, B),语义解释为一个 A (需求任务/状态)涉及一个组织 B ; 4) hasBusi, 描述格式为 hasBusi (A, B),语义解释为一个基本需求任务 A 由一个业务功能 B 实现; 5) hasTask, 描述格式为 hasTask (A, B),语义解释为一个复合需求任务 A 需要分解为一组核心执行模块 B ; 6) hasErd, 描述格式为 hasErd (A, B),语义解释为一个执行模块 A 必须作为一个需求任务 B 进行分析; 7) hasSubR, 描述格式为 hasSubR (A),语义解释为一个资源 A 是一组规模更小的子资源集; 8) hasSubO, 描述格式为 hasSubO (A),语义解释为一个组织 A 是一组规模更小的子组织集; 9) hasLead, 描述格式为 hasLead (A, B),语义解释为领导可管辖普通成员; 10) hasFuncdep, 描述格式为 hasFuncdep (A,

B),语义解释为一个主资源可函数决定一组普通资源; 11) hasPremise, 描述格式为 hasPremise (A, B),语义解释为一个核心执行模块存在必须先执行的一组前置模块; 12) hasCon, 描述格式为 hasCon (A, B),语义解释为业务功能含一前置条件描述; 13) hasInt, 描述格式为 hasInt (A, B),语义解释为业务功能含一个输入描述; 14) hasPro, 描述格式为 hasPro (A, B),语义解释为业务功能有一个过程描述; 15) hasOut, 描述格式为 hasOut (A, B),语义解释为业务功能含一输出描述; 16) hasPost, 描述格式为 hasPost (A, B),语义解释为业务功能必含一后置结果描述; 17) hasType, 描述格式为 hasType (A, B),语义解释为需求任务必有一个任务类型; 18) hasErrand, 描述格式为 hasErrand (A, B),语义解释为角色可参与需求任务; 19) hasOpt, 描述格式为 hasOpt (A, B),语义解释为角色可拥有一组操作使用权; 20) hasAttr, 描述格式为 hasAttr (A, B),语义解释为每一具体操作是在具体 Attribute 上的操作; 21) hasOptTp, 描述格式为 hasOptTp (A, B),语义解释为每一具体操作有一操作类型。ROB 模型本体关系如图3所示。

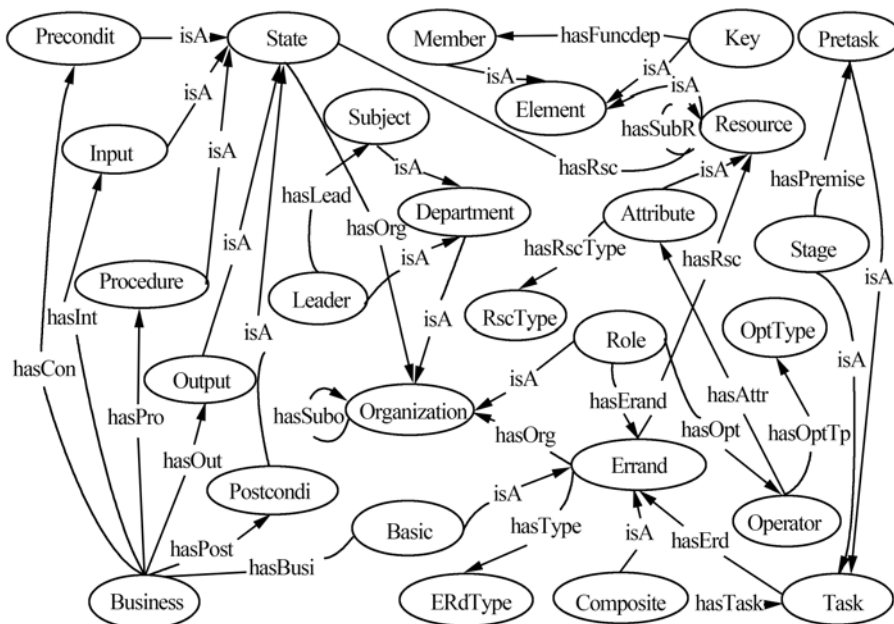


图3 ROB 本体关系

Fig.3 Associations in ROB ontology

3 检验一致的需求建模

3.1 描述逻辑语言

计算机作为智能化信息处理工具,需要将信息或程序赋予某种程度的知识语义,并编码表示;需求富含语义知识,描述逻辑是重要的知识表示工具,用描述逻辑处理需求工程问题是计算智能的一个具体应用。

通过一阶逻辑赋予非逻辑方法以形式语义是描述逻辑最初思想^[10]。本体是对知识显示的概念化描述^[11],而描述逻辑是一种本体描述语言^[12]。描述逻辑建立在概念和关系之上,有很强的语义表达和推理能力,可用于描述需求,检测需求一致性。

ALC 是最小的命题封闭的描述逻辑^[13]语言,在保证推理能力基础上有很强表达能力,近年来有关描述逻辑语言扩展方面的研究也已有许多,如引入构造子、引入具体域^[14-15]。文中采用扩展描述逻辑 SHOIN(D),是通用本体语言 OWL-DL 的逻辑基础。

描述逻辑基本符号包括:1)原子概念: A ;2)概念: C, D ;3)全概念 \top ,空概念 \perp ;4)原子关系: P ;5)关系: R ;6)概念构造子: \neg, \cap, \cup ;7)量词: $\forall R., \exists R.$;8)包含符号: \subseteq, \supseteq ;9)括号: $(,)$;10)个体: a 。根据 ROB 描述需求,文中采用描述逻辑 SHIQ,SHIQ 有如下语法^[16]:① $C ::= A \mid (T \mid (\perp \mid \neg C \mid C \cap D) \mid C \cup D) \mid \forall R. C \mid \exists R. C \mid \theta_n R \mid \theta n R. C \mid \theta n R$;② $R ::= P \mid \neg R \mid R \cap S \mid R \cup S \mid R \circ S \mid R^+ \mid R^-$ 。

描述逻辑知识库包括术语公理库 TBox 和断言事实库 ABox。TBox 公理公式形如 $A \doteq C$ 和 $A \subseteq C$ 。ABox 包括个体断言和关系断言。描述逻辑推理问题包括面向概念的推理和面向个体的推理 2 类。描述逻辑推理的核心问题是可满足性问题,所有推理问题可化为可满足性问题。Tableau 算法是判定可满足性问题有效算法,已得到广泛应用。目前 Tableau 算法扩展、复杂性及优化策略等研究已较为深入且有具体应用。

3.2 ROB 本体知识库

ROB 本体知识库包括概念集合 (Concept)、关系集合 (Association) 和规则集合 (Constraint)。

1) 概念

ROB 本体概念表示类,如基本概念 ERdType 和 RscType 定义为集合: $ERdType ::= \{basic, composite\}$, $RscType ::= \{int, float, char, string, struct, enum, *, OptType ::= \{create, insert, delete, update, select\}$

类含有属性,这些属性与相关类关联,相应概念定义等知识加入到 TBox。如,Errand 具有属性 hasType 关联 ERdType 类,而其子类 Basic 的属性 hasBusi 关联 Business,属性 hasRsc 关联 Resource,属性 hasOrg 关联 Organization。ROB 模型核心概念相应的公式如表 1 所示。

表 1 ROB 概念集合

Table 1 Concept definition in ROB

Concept	Super	Definition
①Errand	T	$Errand ::= T \cap \exists hasType. ERdType \cap \exists hasRsc. Resource \cap \exists hasOrg. Organization \cap (\exists hasBusi. Business \cup \exists hasBusi. Business \cup \exists has a sk.Tasks)$.
②Basic	Errand	$Basic ::= Errand \cap \exists hasBusi. Business$
③Composite	Errand	$Composite ::= Errand \cap \exists hasTask. Tasks$.
④Business	T	$Business ::= \exists hasPro. Procedure \cap (T \cup \exists hasCon. Precondit \cup \exists hasInt. Input \cup \exists hasOut. Output \cup \exists hasPost. Postcondi)$.
⑤Task	T	$Task ::= T \cap \exists hasErd. Errand$
⑥Stage	Task	$Stage ::= T \cap \exists hasPremise. Pre-task$.
⑦State	T	$State ::= T \cap \exists hasRsc. Resource \cap \exists hasOrg. Organization$
⑧Resource	T	$Resource ::= Element \cup Attribute$
⑨Organization	T	$Organization ::= Role \cup Department$
⑩Attribute	Resource	$Attribute ::= Resource \cap \exists hasRscType. RscType$
⑪Role	Organization	$Role ::= Organization \cap \exists hasOpt. Operator \cap \exists hasErrand. Errand$
⑫Operator	T	$Operator ::= T \cap \exists hasAttr. Attribute \cap \exists hasOptTp. OptType$.
⑬Key	Element	$Key ::= Element \cap \exists hasFuncdep. Member$.
⑭Leader	Department	$Leader ::= Department \cap \exists hasLead. Subject$.

2) 关系

ROB 知识库中还以描述逻辑刻画了关系,如关系 hasBusi 描述为: $(\exists hasBusi. Business) \subseteq Basic$,

$\text{Basic} \subseteq (\forall \text{hasBusi. Business})$ 。表示 hasBusi 定义域为 Basic , 而值域为 Business 。表 2 列出主要关系。

表 2 ROB 关系集合

Table 2 Associations in ROB

Associat	Super	Mult	Domain	Range
①hasOrg	T	1:1	Errand \cup State	Organization
②hasStat	T	1:5	Business	State
③hasPost	hasStat	1:1	Business	Postcondi
④hasCon	hasStat	1:1	Business	Precondit
⑤hasInt	hasStat	1:1	Business	Input
⑥hasPro	hasStat	1:1	Business	Procedure
⑦hasOut	hasStat	1:1	Business	Output
⑧hasErd	T	1:1	Task	Errand
⑨hasTask	T	1:N	Composite	Task
⑩has Funcdep	hasMain	1:N	Key	Member
⑪hasPremise	hasMain	1:N	Stage	Pretask
⑫hasLead	hasMain	1:N	Leader	Subject

ROB 知识库还包括概念间的子类关系和关系之间的子类关系, 描述为公理加入到 TBox 中, 描述逻辑中将子类描述为公式, 形如: $\text{Composite} \subseteq \text{Errand}$, $\text{hasPost} \subseteq \text{hasStat}$ 等, 表 3 列出主要子类关系。

表 3 ROB 子类关系集合

Table 3 Included associations in ROB

Included assertion
① $\text{Composite} \subseteq \text{Errand}$, $\text{Basic} \subseteq \text{Errand}$
② $\text{Stage} \subseteq \text{Task}$, $\text{Pretask} \subseteq \text{Task}$
③ $\text{Leader} \subseteq \text{Department}$, $\text{Subject} \subseteq \text{Department}$, $\text{Role} \subseteq \text{Organization}$, $\text{Department} \subseteq \text{Organization}$
④ $\text{Postcondi} \subseteq \text{State}$, $\text{Precondit} \subseteq \text{State}$, $\text{Input} \subseteq \text{State}$, $\text{Procedure} \subseteq \text{State}$, $\text{Output} \subseteq \text{State}$
⑤ $\text{Key} \subseteq \text{Element}$, $\text{Member} \subseteq \text{Element}$, $\text{Attribute} \subseteq \text{Resource}$, $\text{Element} \subseteq \text{Resource}$
⑥ $\text{hasPost} \subseteq \text{hasStat}$, $\text{hasCon} \subseteq \text{hasStat}$, $\text{hasInt} \subseteq \text{hasStat}$, $\text{hasPro} \subseteq \text{hasStat}$, $\text{hasOut} \subseteq \text{hasStat}$

3) 约束

ROB 模型概念和关系必须满足一组约束, 这些约束用于保证需求完整性和一致性。按约束来源可分为概念约束、关系约束和实例约束。

(1) 概念约束, 限定实例必须的数量, 保证概念完整, 如: 每个需求任务必有一个资源、一个组织和

一个功能实现等, 表 4 列出主要的概念约束。

表 4 ROB 概念约束

Table 4 Numbers constraint on ROB concept

Numbers Constraint on Concept
① $\text{Errand} \subseteq \geq 1 \text{hasRsc} \cap \leq 1 \text{hasRsc}$
② $\text{Errand} \subseteq \geq 1 \text{hasOrg} \cap \leq 1 \text{hasOrg}$
③ $\text{Basic} \subseteq \geq 1 \text{hasBusi} \cap \leq 1 \text{hasBusi}$
④ $\text{Business} \subseteq \geq 0 \text{hasPost} \cap \leq 1 \text{hasPost}$
⑤ $\text{Business} \subseteq \geq 0 \text{hasInt} \cap \leq 1 \text{hasInt}$
⑥ $\text{Business} \subseteq \geq 1 \text{hasPro} \cap \leq 1 \text{hasPro}$
⑦ $\text{Business} \subseteq \geq 0 \text{hasOut} \cap \leq 1 \text{hasOut}$
⑧ $\text{Business} \subseteq \geq 0 \text{hasCon} \cap \leq 1 \text{hasCon}$
⑨ $\text{Composite} \subseteq \geq 2 \text{hasTask}$
⑩ $\text{Task} \subseteq \geq 1 \text{hasErd} \cap \leq 1 \text{hasErd}$
⑪ $\text{Leader} \subseteq \geq 0 \text{hasLead} \cap \leq N \text{hasLead}$
⑫ $\text{Stage} \subseteq \geq 0 \text{hasPremise} \cap \leq N \text{hasPremise}$
⑬ $\text{State} \subseteq \geq 1 \text{hasRsc} \cap \leq 1 \text{hasRsc}$
⑭ $\text{State} \subseteq \geq 1 \text{hasOrg} \cap \leq 1 \text{hasOrg}$
⑮ $\text{Key} \subseteq \geq 0 \text{hasFuncdep} \cap \leq N \text{has Funcdep}$

(2) 关系约束, 限定概念必要的属性, 表示概念必须存在一些关系, 如: 每个需求任务必对应一组执行模块, 对应一个表示功能过程的状态等; 表 5 列出了主要关系约束。

表 5 ROB 关系约束

Table 5 Relation constraints in ROB

Relation Constraint
① $\text{Operator} \subseteq \forall \text{hasAttr} (\text{Attribute} \cap \exists \text{hasRscType})$
② $\text{Basic} \subseteq \forall \text{hasBusi} (\text{Business} \cap \exists \text{hasPro})$
③ $\text{Role} \subseteq \forall \text{hasOpt} (\text{Operator} \cap \exists \text{hasOptTp})$
④ $\text{Task} \subseteq \forall \text{hasErd} (\text{Errand} \cap \exists \text{hasType})$
⑤ $\text{Task} \subseteq \forall \text{hasErd} (\text{Errand} \cap \exists \text{hasRsc})$
⑥ $\text{Task} \subseteq \forall \text{hasErd} (\text{Errand} \cap \exists \text{hasOrg})$
⑦ $\text{Business} \subseteq \forall \text{hasStat} (\text{State} \cap \exists \text{hasRsc})$
⑧ $\text{Business} \subseteq \forall \text{hasStat} (\text{State} \cap \exists \text{hasOrg})$

(3) 实例约束, 通过限定实例间的对应关系, 保证语义完整和一致性, 如: 若任务 a 对应组织 b 和资源 c , 而 b 对应资源 d , 则 d 与 c 是相同实例。表 6 列

出主要的实例约束规则。

表 6 ROB 实例约束

Table 6 Individual constraints in ROB

Individual Constraint	
①	$\text{hasErrand}(\text{?}e, \text{?}x) \wedge \text{hasOrg}(\text{?}x, \text{?}z) \rightarrow \text{hasSubO}(\text{?}e, \text{?}z)$
②	$\text{hasSubO}(\text{?}x, \text{?}o) \rightarrow \text{hasLead}(\text{?}x, \text{?}o)$
③	$\text{hasRsc}(\text{?}x, \text{?}y) \wedge \text{hasBusi}(\text{?}x, \text{?}z) \wedge \text{hasState}(\text{?}z, \text{?}s) \wedge \text{hasOrg}(\text{?}s, \text{?}o) \rightarrow \text{hasRscR}(\text{?}x, \text{?}o)$
④	$\text{hasOrg}(\text{?}x, \text{?}y) \wedge \text{hasBusi}(\text{?}x, \text{?}z) \wedge \text{hasState}(\text{?}z, \text{?}s) \wedge \text{hasOrg}(\text{?}s, \text{?}o) \rightarrow \text{hasSubO}(\text{?}x, \text{?}o)$
⑤	$\text{hasTask}(\text{?}x, \text{?}y) \wedge \text{hasErd}(\text{?}y, \text{?}z) \rightarrow \text{isA}(\text{?}x, \text{?}z)$
⑥	$\text{hasErd}(\text{?}x, \text{?}y) \wedge \text{hasTask}(\text{?}y, \text{?}z) \rightarrow \text{isA}(\text{?}x, \text{?}z)$

4) 性质

通过限定实例间关系和性质,建立可推导的公式,约束对应的需求语义冲突,如:hasSub 与 isA 存在互逆性质;ROB 概念、实例间所有性质包括传递性(Transitive)、对称性(Symmetry)、函数关系(Function)和逆关系(inverse)、属性继承(Property Inherit)和特性继承(Characteristics inherit)。表 7 列出 ROB 性质。

表 7 ROB 性质

Table 7 Universal inference-rule in ROB

	regular	Associate
Transitive	$\text{hasSubO}, \text{hasSubR}, \text{hasSub}, \text{isA}, \text{hasLead}, \text{hasFuncdep}, \text{hasPremise},$	
Symmetry	$\text{differentFrom}, \text{sameAs}$	
Function	$\text{hasBusi}, \text{hasOrg}, \text{hasRsc}, \text{hasErd}$	
inverse	$(\text{hasSub}, \text{isA}), (\text{hasErd}, \text{has})$	
Property Inherit	$\text{hasSub}, \text{isA}, \text{InstancOf}$	
Characteristics inherit	$(\text{hasLead}, \text{hasSubO}), (\text{hasFuncdep}, \text{hasSubR}), (\text{hasSubO}, \text{hasSub}), (\text{hasSubR}, \text{hasSub})$	

根据关系性质可构造出更多推导公式,用于推理验证,以检查和判定实例断言可满足性。

3.3 检验一致的需求建模过程

ROB 本体知识库包括概念集合(Concept)、关系集合(Association)和规则集合(Constraint)。

定义 1 ROB 知识库。ROB 知识库 $KB = \{TBox, ABox\}$, 其中 TBox 术语公理集合包含了 ROB 概念(表 1)、关系(表 2、3)、约束(表 4~6)公式; ABox 是 ROB 断言模块的有限集合。

定义 2 ROB 断言模块。ROB 断言模块是 ROB 断言的有限集合,包括概念断言和角色断言。

如:一个 ROB 断言模块(AssertionBlock) ablock 含一组断言:概念断言 Errand(?e), 表示 e 是一个需求任务; Business(?b), 表示 b 是一个业务功能; Organization(?o), 表示 o 是一个组织; 关系断言 hasSubO(?o, ?o₁), 表示组织 o 中包含规模更小的组织 o₁。

通过定义 ROB 本体概念体系以及约束、性质,在描述逻辑中建立知识库,并将需求建模过程描述为创建 ROB 断言模块的过程,利用描述逻辑推理能力进行一致性检查和验证,目的是发现不完整、不一致问题。

性质 1 完整性。一个 ROB 断言模块若满足所有概念约束则称为具有概念完整性;满足所有关系约束称具有关系完整性;满足所有实例约束称具有实例完整性。一个 ROB 断言模块具有完整性当且仅当同时具有概念完整性、关系完整性和实例完整性。

定义 3 冲突。对一个 ROB 知识库 KB K, ROB 断言模块中任意断言按 ROB 性质推导公式,若出现形如 $\neg R(a, b) \wedge R(a, b)$ 的推论,即 ROB 断言模块冲突。

性质 2 一致性。若一个 ROB 断言模块具有完整性,且对知识库 KB K 不存在冲突,则称具有一致性。

ROB 需求一致的建模过程是在本体知识库基础上不断扩展断言知识的过程,下面进行说明:

第一,扩展断言知识过程描述:1) 创建 ROB 断言模块,捕获需求,根据语义抽取 ROB 需求对象及联系,将这些对象、联系表示为 ROB 断言,形成 ROB 断言模块;2) 完整性检查,若具有完整性则进行第 3 步,否则根据检查结果增加断言以扩展 ROB 断言模块,扩展后重新执行第 2 步完整性检查;3) 一致性检查,若具有一致性则进行第 4 步,否则根据检查结果修改断言后执行第 2 步完整性检查;4) 扩展断言知识。

ROB 断言模块对应一个在知识库 KB 上的解释(即 ROB 断言模块的一个模型) M_1 , M_1 作为新的断言知识扩展为特定断言库 ABox A_1 ; ROB 断言模块具有原子性,一个 ROB 断言模块要么被取消,要么作为断言知识增加到 ABox。

第二,扩展断言库过程描述:ROB 基于递归分解思想反复获取需求,每次增加新的 ROB 断言模块都对应一模型 M , 整个建模过程就是一个有限集合: $\{M_i \mid i = 1, 2, \dots, n\}$, 其中每个元素 M_i 是一个模型;模型 M_i 对应特定的断言库 ABox A_i , 可作为

模型 $M_i + 1$ 的 ABox, 与公理知识库 TBox T 一起构成 $M_i + 1$ 的完整知识库。

ROB 建模过程不断创建 ROB 断言模块, 经过完整性、一致性检验后转化为断言知识增加到断言知识库, 核心工作就是完整性、一致性检验。由性质 2 可知, 一致性问题包含完整性问题; 一致性检验是在 ROB 知识库基础上的推理问题, 可转化为可满足问题; 可满足问题推理验证 Tableau 算法等可参阅文献[17-20], 文中主要提出 ROB 需求一致的建模方法, Tableau 算法的扩展、复杂性和优化策略等。

4 案例分析

需求建模是一项浩大工程, 常需反复迭代。文中在论述理论的基础上编制了相应的建模分析软件, 提高了 ROB 建模自动化与智能化水平, 本节以一个案例展示 ROB 需求建模过程。案例: 某学校研发智能化综合实训管理平台的需求。

4.1 扩展断言知识

ROB 知识库中初始 ABox A 为空, 即 $A_{Box} A_0 = \emptyset$ 。

1) 创建 ROB 断言模块

(1) 捕获需求: ①项目名称: 智能化综合实训管理平台; ②需求单位: 某学校; ③目标任务: 实训教学教务综合管理; ④项目目的: 提高实训教学质量, 提高实训室管理水平。(2) 抽取 ROB 需求对象个体: t_0 : 实训室、实训教学综合管理; b_0 : 智能化综合实训管理平台; o_0 : 某学校; r_0 : 整体资源; (3) 创建 ROB 断言模块: $ablock0 = \{ Task(t_0), Basic(e_0), Business(b_0), Organization(o_0), Resource(r_0), hasErand(t_0, e_0), hasBusi(e_0, b_0), hasOrg(e_0, o_0), hasRsc(e_0, r_0) \}$

2) 完整性检查

遍历 ROB 断言模块中断言, 应用 Tableau 算法依次进行完整性检查, 完整性可满足性问题约束包括: (1) 概念完整性约束; (2) 关系完整性约束; (3) 实例完整性约束。

例如: 遍历 ROB 断言模块 $ablock0$ 中个体断言, 检验对概念约束的可满足性, 由于不满足 $Business \sqsubseteq \geq 1 hasPro \sqcap \leq 1 hasPro$ (表 4 公式⑥), $ablock0$ 不具有完整性, 添加个体断言 $Procedure(p_0)$ 以扩展 $ablock0$, 即: $ablock0 = ablock0 \cup \{ Procedure(p_0) \}$, 然后重新执行完整性检查。

3) 一致性检查

遍历 ROB 断言模块中断言, 应用 Tableau 算法依次判定可满足性: (1) 把 TBox 带入到 ABox 中消除 TBox; (2) ABox 中断言转化为否定正则形; (3)

ABox 上应用扩展规则直到无可用规则; (4) 只要存在解释, 也就是说只要有一个 ABox 一致, 则原始 ABox 一致。

4) 扩展断言知识

将 ROB 断言模块中的所有断言公式添加到 ABox A0 中, 如: $A_0 = A_0 \cup ablock0$; $ablock0 = \{ Task(t_0), Basic(e_0), Business(b_0), Organization(o_0), Resource(r_0), Procedure(p_0), hasErand(t_0, e_0), hasBusi(e_0, b_0), hasOrg(e_0, o_0), hasRsc(e_0, r_0), hasOrg(p_0, o_0), hasRsc(p_0, r_0) \}$ 。

4.2 扩展断言库

1) ROB 需求分解

ROB 建模基本方法是分解, 每一次分解都对应一组 ROB 断言模块: (1) 分解得新需求转化为 ROB 需求断言模块; (2) 分解本身需转化为 ROB 断言模块。对每一个 ROB 断言模块, 都需要经过完整扩展断言知识(4.1)的过程。

例如, 按 ROB 结构对需求任务 e_0 分解为: t_1 (实训教学核心任务); t_2 (实训教学基础任务); t_3 (实训教学日常任务)。(1) 分解得新的需求: ① $Task(t_1)$; ② $Task(t_2)$; ③ $Task(t_3)$ 。分别执行完整的扩展断言知识的过程; 分别对应为有序的断言知识库: ① $ABox A_1$; ② $ABox A_2$; ③ $ABox A_3$ 。(2) 分解本身是断言知识: ① 对应的 ROB 断言模块 $ablock(decomposed) = (KB - \{ Basic(e_0) \}) \cup \{ Composite(e_0) \}$; ② 完整性检查, 得: $ablock(decomposed) = ablock(decomposed) \cup \{ hasTask(e_0, t_1), hasTask(e_0, t_2), hasTask(e_0, t_3) \} \cup \{ hasPremise(t_1, t_2) \}$; ③ $ablock(decomposed)$ 经过一致性检查和扩展断言知识后, 对应新的断言知识库 $ABox Ade1$ 。

分解说明: ①需求分解逐步进行, 细化的程度由分析人员确定; ②分解的结果作为新需求, 扩展为断言知识的过程是相同的; ③重点讨论保证一致地建模过程和方法, 为论述方便, 这里不对详细分解过程描述。

2) 扩展断言库

ROB 建模过程逐步分解细化需求, 不断产生 ROB 断言模块, 每一个 ROB 断言模块经过一致性检验和扩展, 作为断言知识增加到知识库中, 在初始知识库基础上形成一个新的 ABox。从最初对应原始知识库 $ABox = \emptyset$, 到分析结束整个过程, 形成一个有序的 ABox 有限集合: $ABox Aset = \{ ABox A_i \mid i = 1, 2, \dots, n \}$, 其中每个元素是一个特定的断言知识库, Aset 记录了断言知识库的扩展过程, 表示了需求分析的整个过程, 为基于推理的需求管理提供了方

便。例如,案例需求分解过程对应一个有序 ABox 的集合,断言库扩展过程如图 4 所示。

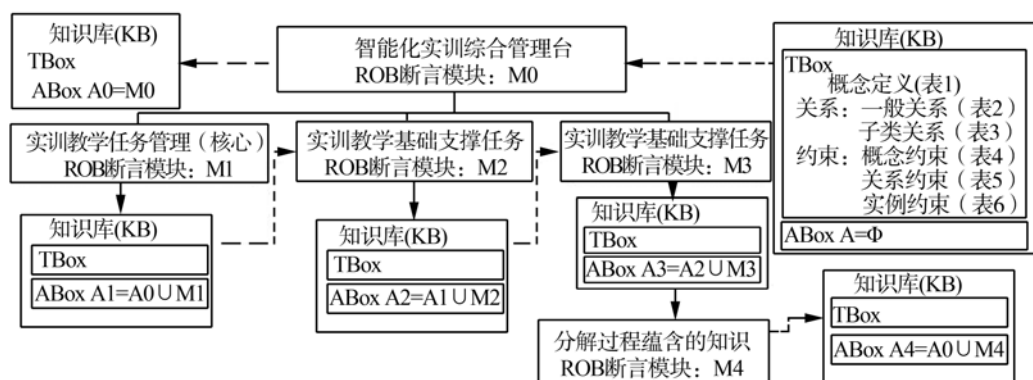


图 4 需求任务分解案例断言库扩展过程

Fig.4 Expansion process of ROB assertion knowledge

断言知识库扩展过程每一步通过断言知识扩展实现,断言知识扩展需经过完整性检查、一致性检查,保证了完整一致地进行需求分析。

5 结束语

ROB 需求建模提取需求分析要素:资源、组织和业务,从这 3 个视点抽取需求信息,通过演绎分解建立需求树,分析约束规则消除冗余,将不同视点下的需求树装配为完整需求树。ROB 模型引入本体理论,建立概念层次体系,形成概念及关系,进一步结合语义建立了约束与规则,基于描述逻辑构建了需求模型公理知识库,根据推理规则及公式性质建立了基本的断言知识库,ROB 需求建模过程不断建立新的实例断言,将需求建模过程转化为断言知识库的逐步扩展过程。由于将需求完整问题转化为需求一致问题,需求一致检验归结为规则约束下的可满足问题,利用现有研究成果 Tableau 算法等检验,保证了 ROB 建模的需求一致性。

检验一致的 ROB 需求建模方法借鉴了现有的需求建模技术和观点,设计了一组规则用于推理、验证和约束,并基于逻辑知识库的推理能力实现需求模型的一致性检验。目前研究存在一些不足:1) 主要关注功能需求而对非功能需求等其他需求问题考虑不足;2) 着重于特定需求建模实例和复杂问题的解决,没有从软件工程全局设计模型;3) 专注于描述逻辑知识库中逻辑推理问题而对 ROB 需求分解方法缺乏规范指导,依然依赖于专家经验;4) 由于将需求一致性问题转化为可满足问题,而 Tableau 算法复杂难解。

进一步工作主要有:1) 设计合理的实验,通过

实验分析和数据比对,发现实际可行的有效算法;2) 通过专家审议和验证分析,规范 ROB 需求分解过程和评估度量体系,以进一步提高建模的自动化程度;3) 分析运用非功能需求等其他需求的建模研究成果,从功能需求、非功能需求等扩展 ROB 模型,形成全面、统一的建模方法;4) 分析软件全生命周期的工作,基于 ROB 在软件工程各阶段设计相应的工作,以形成完整的需求建模体系,进一步提高软件工程的自动化智能化水平。

参考文献:

- [1] SPANOUDAKIS G, ZISMAN A. Inconsistency management in software engineering: survey and open research issues [M]. Handbook of Software Engineering and Knowledge Engineering. Singapore: World Scientific Publishing Co., 2001: 329-380.
 - [2] SPANOUDAKIS G, FINKELSTEIN A, TILL D. Overlaps in requirements engineering[J]. Automated Software Engineering, 1999, 6(2): 171-198.
 - [3] NUSEIBEH B, EASTERBROOK S, RUSSO A. Leveraging inconsistency in software development [J]. Computer, 2000, 33(4): 24-29.
 - [4] NUSEIBEH B, FINKELSTEIN A, KRAMER J. Viewpoints: meaningful relationships are difficult! [C] // Proc of the 25th Int'l Conf. on Software Engineering. Portland: IEEE Computer Press, 2003: 676-683.
 - [5] 王智学,董庆超,朱卫星,等.一种支持能力需求概念一致性和合理性分析的方法[J].计算机学报,2013,36(1): 10-21.
- WANG Zhixue, DONG Qingchao, ZHU Weixing. An approach for conceptual analysis on capability requirements

- consistency and reasonability [J]. Chinese Journal of Computers, 2013, 36(1): 10-21.
- [6] NECHES R, FIKES R E, FININ T, et al. Enabling technology for knowledge sharing[J]. AI Magazine, 1991, 12(3): 36-56.
- [7] GRUBER T R. A translation approach to portable ontology specifications[J]. Knowledge Acquisition, 1993, 5(3): 199-220.
- [8] 陆汝铃, 金芝, 陈刚. 面向本体的需求分析[J]. 软件学报, 2000, 11(8): 1009-1017.
- LU Ruqian, JIN Zhi, CHEN Gang. Ontology-oriented requirements analysis[J]. Journal of Software, 2000, 11(8): 1009-1017.
- [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[J]. 软件学报, 2011, 22(2): 177-194.
- CHEN Xiaohong, YIN Bin, JIN Zhi. Ontology-guided requirements modeling based on problem frames approach[J]. Journal of Software, 2011, 22(2): 177-194.
- [10] FRANZ B, DIEGO C, DEBORAH L, et al. The description logic handbook: theory, implementation, and applications[M]. New York: Cambridge University Press, 2003: 555-574.
- [11] GRUBER T R. A translation approach to portable ontology specifications[J]. Knowledge Acquisition, 1993, 5(2): 199-220.
- [12] FRANZ B, IAN H, ULRIKE S. Description logics as ontology languages for the semantic web[C]// Mechanizing Mathematical Reasoning: Essays in Honor of JORG H Siekmann on the Occasion of His 60th Birthday, 2005: 228-248.
- [13] SCHMIDT S, SMOLKA G. Attributive concept descriptions with complements [J]. Artificial Intelligence, 1991, 48(1): 1-26.
- [14] 常亮, 史忠植, 陈立民, 等. 一类扩展的动态描述逻辑[J]. 软件学报, 2010, 21(1): 1-13.
- CHANG Liang, SHI Zhongzhi, CHEN Limin, et al. Family of extended dynamic description logics[J]. Journal of Software, 2010, 21(1): 1-13.
- [15] 常亮, 史忠植, 古天龙, 等. 可判定的时序动态描述逻辑[J]. 软件学报, 2011, 22(7): 1524-1537.
- CHANG Liang, SHI Zhongzhi, GU Tianlong, et al. Decidable Temporal Dynamic Description Logics[J]. Journal of Software, 2011, 22(7): 1524-1537.
- [16] 沈国华, 张伟, 黄志球, 等. 基于描述逻辑的特征语义建模及验证[J]. 计算机研究与发展, 2013, 50(7): 1501-1512.
- SHEN Guohua, ZHANG Wei, HUANG Zhiqiu. Description-logic-based feature modeling and verification [J]. Journal of Computer Research and Development, 2013, 50(7): 1501-1512.
- [17] 刘全, 崔志明, 高阳, 等. 一种逻辑强化学习的 tableau 推理方法[J]. 智能系统学报, 2008, 3(4): 355-360.
- LIU Quan, CUI Zhi-ming, GAO Yang, et al. Tableau reasoning method based on logical reinforcement learning[J]. CAAI Transactions on Intelligent Systems, 2008, 3(4): 355-360.
- [18] 叶育鑫, 欧阳丹彤, 刘瑶, 等. 基于 SHOIQ(D) 的本体一致性检测[J]. 计算机工程与科学, 2009, 31(8): 7-10.
- YE Yuxin, OUYANG Dantong, LIU Yao, et al. Consistency checking of the SHOIQ(D)-based ontology[J]. Computer Engineering & Science, 2009, 31(8): 7-10.
- [19] 石莲, 孙吉贵. 描述逻辑综述[J]. 计算机科学, 2006, 33(1): 194-197.
- SHI Lian, SUN Jigui. Description Logic Survey [J]. Computer Science, 2006, 33(1): 194-197.
- [20] 古华茂, 王勋, 凌云, 等. 完全析取范式群判定 SHOIN(D)-可满足性[J]. 软件学报, 2010, 21(8): 1863-1877.
- GU Huamao, WANG Xun, LING Yun, et al. Determining the SHOIN(D)-satisfiability with a complete disjunctive normal form group [J]. Journal of Software, 2010, 21(8): 1863-1877.

作者简介:



郭新峰, 男, 1974 年生, 硕士, 主要研究方向软件工程、智能计算及应用。



马世龙, 男, 1953 年生, 教授, 博士生导师, 博士, 主要研究方向为计算机软件与理论。



吕江花, 女, 1975 年生, 副教授, 博士后, 主要研究方向为计算机软件与理论。