

一种面向方面需求模型的分析与验证方法

金 英 刘华斌 张 鹏

(吉林大学计算机科学与技术学院 长春 130012)

摘 要 建立具有完整性和一致性的需求模型是软件项目成功的基础,验证需求模型是否具有相关性在软件开发中具有重要的意义.文中针对面向方面文档驱动需求方法中的需求分析过程,给出了关注点建模过程中的若干验证策略,提出了基于本体的关注点模型验证的相关方法,并通过一个实例说明了文中方法的有效性.

关键词 面向方面需求分析;关注点关联图;本体;一致性验证

中图法分类号 TP311 DOI号 10.3724/SP.J.1016.2013.00063

An Approach to Analysing and Verifying Aspect-Oriented Requirements Model

JIN Ying LIU Hua-Xiao ZHANG Peng

(College of Computer Science and Technology, Jilin University, Changchun 130012)

Abstract Establishing complete and consistent requirements model is the basis for the success of software projects, and verifying requirements models plays key role in software development. Under the framework of aspect oriented document driven requirements engineering, several verifying schema are proposed for the process of modeling concerns, and an ontology based approach to verifying concerns models is presented, a case study is made for indicating the effectiveness of our approach.

Keywords aspect-oriented requirements analysis; concerns' relation graph; ontology; consistency verify

1 引 言

需求工程在软件开发过程中具有方向性和决策性的意义,需求建模是软件需求工程中的一个重要阶段,一个定义良好的需求模型是保证软件项目成功的前提之一.需求建模通过切实有效的方法和技术对所获取需求中的行为、特征和相关约束进行抽取和精化,构建一个准确描述软件系统功能和约束的需求模型.随着软件系统规模的不断扩大,软件需求模型也越来越复杂,如何验证需求模型的完整性、一致性等性质是需求工程中的一个挑战.

目前有很多需求建模方法^[1-5]和需求模型验证方法^[6-8],不同的需求建模方法从不同的视角看待软件系统,体现了不同的建模理念.当前在学术界受到广泛关注的需求建模方法包括面向对象方法^[5]、面向方面(Asspect)的方法^[3]、面向特征(Feature)的方法^[4]和面向目标(Goal)的方法^[2].针对不同的需求模型有不同的模型验证方法,主要有基于语义分析的验证方法^[9]、基于状态机推理的验证方法^[6]、基于本体约束验证方法^[10-11]等,从不同侧面一定程度上解决需求验证问题.

在文献[12]中提出了一种面向方面文档驱动需求方法,基于该方法框架,本文给出了一种针对面向

收稿日期:2012-06-25;最终修改稿收到日期:2012-11-24. 本课题得到新世纪优秀人才支持计划(NCET-10-0436)、吉林省自然科学基金项目(20101509)、中央高校基本科研业务费专项基金(200903193,201103133)资助. 金 英,女,1971年生,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为需求工程、软件形式化. E-mail: jinying@jlu.edu.cn. 刘华斌(通信作者),男,1986年生,博士研究生,主要研究方向为软件形式化、软件需求工程. E-mail: jlulhx@gmail.com. 张 鹏,男,1986年生,博士研究生,主要研究方向为语义网、本体工程.

方面需求分析模型的验证方法,给出了获取和规范描述关注点模型过程中的验证策略,提出了基于本体的关注点模型验证方法,归纳、定义并实现了基于本体的检验面向方面需求分析模型的完整性和一致性等性质的若干约束条件,最后,通过一个实例说明了本文所提出方法的有效性。

本文第 2 节概要介绍面向方面文档驱动需求工程方法的主要过程以及关注点、关注点关联关系和关注点关联图等定义;第 3 节给出获取和规范描述关注点模型的方法以及在该过程中的验证策略;第 4 节阐述基于本体的需求模型定义和验证方法;第 5 节通过实例分析验证本文方法的可行性和有效性;第 6 节针对国内外相关工作做了比较;最后总结全文并提出下一步工作设想。

2 面向方面需求工程概述

关注点(Concern)指的是一个特定的目标、概念或者兴趣域.一个关注点通常对应一个或者多个软件相关人员所关心的问题或特性,通常被定义为一组相互关联的需求集合.“关注点分离(Separate of Concern)”^[13]是软件工程中应遵守的必要原则,以分离关注点为核心的面向方面需求工程 AORE^[3]在需求阶段引入了关注点的概念,有效降低了系统的复杂性和耦合性,提高了需求的模块化程度.文献[12]中提出的面向方面文档驱动需求分析方法,将建立形式化需求文本的文档驱动方法与面向方面需求工程相结合,设计了一个从需求文本出发,在识别、定义关注点及关注点关联图的基础上,定义需求的形式化文档的逐步精化的过程.本文将在此工作的基础上,给出一个关注点定义及关注点模型完整性和一致性的验证方法.图 1 中介绍了面向方面文档驱动需求方法的主要过程。

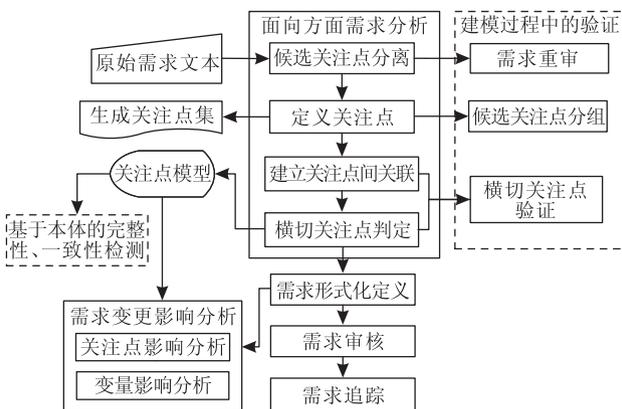


图 1 面向方面文档驱动需求方法的主要过程

如图 1 所示,面向方面文档驱动需求分析方法首先根据面向方面需求分析原则,对原有的文本需求进行分离关注点、定义关注点、定义关注点间关系、判定横切关注点来构建关注点模型,此后对需求进行形式化描述,并进行需求变更影响分析,最后重新审视变量与需求间的关系,定义需求的可追踪性.本文在原有工作基础上,针对其中的面向方面需求分析方法部分给出了在关注点建模过程中的验证策略,以及基于本体的关注点模型验证方法,如图 1 中的虚线部分所示。

下面给出关注点、关注点依赖关系、关注点关联图的相关定义。

定义 1. 关注点. 一个关注点 C 可以定义为一个四元组 $(id, Des, Rs, type)$, 其中 id 为关注点 C 的名字, Des 为关注点 C 的文本描述, Rs 指与关注点 C 相关联的需求集合, $type$ 指的是关注点的类型. 关注点一共分成 4 种类型, 分别是: $topic$ 表示主题关注点; $kinds$ 表示功能、行为和状态类型的关注点; $instance$ 表示更为具体的 $kinds$ 类型关注点; $property$ 表示 $kinds/instance$ 类型关注点的特征。

定义 2. 关注点关联关系. 一个关注点关联关系 RL 可以定义为一个五元组 $(C_i, C_j, reqLs, type, crosscutting)$, 表示从关注点 C_i 通过需求集合 $reqLs$ 关联到关注点 C_j , 并且 $reqLs \in C_i.Rs \cap C_j.Rs$. 关联关系一共分成 6 种类型, 分别是 $relates_to$, 描述一个 $topic$ 类型关注点到 $kinds$ 类型关注点的关联; $kinds_of$, 描述一个 $kind$ 类型关注点到另一个 $kind$ 类型关注点的关联; $instance_of$, 描述一个 $kinds$ 类型关注点到一个 $instance$ 类型关注点的关联; $applies_of$, 描述一个 $instance$ 类型关注点到一个 $property$ 类型关注点的关联; $dependent$, 没有定义域和值域的限制, 描述逻辑上的关联关系; $motivate$ 与 $dependent$ 为互逆关系. $crosscutting$ 的值为 $Bool$ 类型, 当其值为真时代表 C_i, C_j 之间存在横切关系, C_i 为 C_j 的横切关注点, 否则 C_i, C_j 间不存在横切关系。

此处关于关注点关联关系的定义在文献[12]的基础上增加了一个 $Bool$ 元素来表示关注点的横切关系。

定义 3. 关注点关联图. 设存在一个关注点集合 C_{set} 、存在一个关注点关联关系集合 RL_{set} , 对于 $\forall C_i \forall C_j \in C_{set}$, 若存在 C_i 与 C_j 间的依赖关系, 则认为 C_i 与 C_j 之间存在一条有向边. 以 C_{set} 作为点集, 以 RL_{set} 作为边集组成的图 G , 叫做关注点关联图。

3 关注点模型建立过程中的验证

3.1 关注点模型的获取方法

在文献[14]中指出,一个系统的关注点可以被归纳为 3 个层次,如图 2^[14]所示。

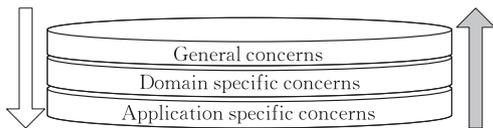


图 2 关注点的层次划分

第 1 层为通用关注点,表示系统中常见模块,例如安全控制等;第 2 层为领域关注点,表示通过领域分析获得的关注点,例如对于实时系统来说,时间控制就是一个领域关注点;第 3 层为实现系统关注点,表示完成需求所要实现的系统功能. 关注点模型获取方法主要分成以下几个步骤:

(1) 对于原始的需求文本进行按条目的编号,构建一个型如 R_1, R_2, \dots, R_n 的需求集 R .

(2) 自顶向下分析. 这是一个精炼化的过程,首先建立通用关注点,然后建立领域关注点,并构建其与通用关注点间的关系,之后通过分析需求集 R ,建立更多的具体化的关注点.

(3) 自底向上分析. 使用词法分析工具,以需求集 R 作为原始输入,对需求进行词法分析以获取需求中所描述的动作、行为、特征、目标、功能等关键词,并称之为候选关注点集 CC_{set} . 然后从候选关注点集 CC_{set} 中最基层的候选关注点出发,通过对候选关注点进行分组,将其中比较独立和重要的候选关注点表示成关注点. 对于一些可以被其它关注点所包含的候选关注点,将其归并进相关的关注点,这样一步步自底向上的建立关注点模型.

(4) 结合. 将自顶向下分析与自底向上分析所构建的模型进行结合,并结合实际分析,获取最终的关注点模型.

(5) 横切关注点判定. 判定一个关注点 C_i 是否为横切关注点的主要依据是,观察依赖于该关注点的某一需求 R_j 是否同时依赖于其它关注点集 C_{set} , 若 R_j 可以被重新定义分割成几条新的需求并且可以被 C_{set} 中的关注点所分别实现,则对 R_j 进行重新定义,若 R_j 不可分割且关注点集 C_{set} 为了实现需求 R_j 需要依赖 C_i 中的功能,则将 C_{set} 中与 R_j 相关的共同功能提取出来,加入到 C_i 关注点中,此时关注点 C_i 即为一个横切关注点集 C_{set} 的横切关注点.

经过以上的 5 个主要步骤,即可形成一个面向方面需求分析的关注点模型,模型中包括需求集 R 、关注点集 C 、关注点关联关系集 RL 、横切关注点集 C_{cut} . 然而在这个过程中,存在着以大量手工操作和开发者经验作为依据的分析过程,为了验证模型的准确性,本文给出了一系列保证关注点模型正确性的方法.

3.2 建模过程中的验证策略

本文主要基于 3 个方面,在模型获取过程中开展验证工作,分别是需求依赖的准确性验证、候选关注点分组的合理性检验、横切关注点必要条件判定.

3.2.1 需求依赖准确性的验证策略

建立关注点模型的实际过程中,某需求 R_i 与任何关注点存在依赖主要存在两种可能. 其一,对某需求 R_i 的词法分析并未获得相关候选关注点,实际上一种可能是 R_i 中将其具体需求描述的比较模糊,并没有清晰地指出它所对应的候选关注点;另一种可能是 R_i 中本身不包含需要系统完成的实际功能. 其二,在自底向上的构建关注点模型的过程中,缺少对需求与动作间依赖关系的重定向. 为了解决上述问题,本文给出了相关的验证策略.

需求重审策略: 在建模开始阶段,若需求 R_i 中不包含任何候选关注点,则对其进行重审,若 R_i 描述中隐含了其实际关联候选关注点 CC_j , 则建立 R_i 到 CC_j 的依赖关系,并通过与需求提供人员进行沟通,修改需求 R_i 以使其描述的更为直观;若 R_i 中所描述的内容,对实际开发并无影响,即不需要某一候选关注点 CC_k 来实现该需求,则忽略该需求.

候选关注点分组中的需求依赖重定向策略: 建模过程中的候选关注点分组,实际上是将原有的相互关联一个候选关注点集 CC_{set} 重组成为一个新的候选关注点 CC_i , 并将 CC_{set} 中的子元素表示成 CC_i 的子候选关注点,分组过程结束后所剩余的单独候选关注点和候选关注点组就被定义成关注点. 若 $CC_j \in CC_{set}$, 则依赖于候选关注点 CC_j 的需求 R_n 需要对其依赖关系进行重定向,删去 R_n 与 CC_j 之间的依赖关系,建立 R_n 与 CC_i 之间的依赖关系.

3.2.2 候选关注点分组验证策略

前文中已经简要描述了自底向上构建关注点模型的过程,在对需求集进行词法分析之后,会获得大量的动作、行为、目标、功能等候选关注点,而其中的很多候选关注点不足以单独作为关注点存在,因此需要进行候选关注点分组以组建关注点. 如果不同候选关注点与相同的需求之间存在关联,则这些候

选关注点可能会被分组, 候选关注点分组是一个迭代进行的过程, 多个候选关注点所组成的候选关注点组也将被看成是一个独立的新候选关注点, 并参与到下一阶段的候选关注点分组, 直到所有候选关注点组和独立候选关注点之间已经不存在隶属关系, 此时每一个候选关注点组和独立的候选关注点都被定义为一个关注点. 候选关注点分组示例如图 3 所示.

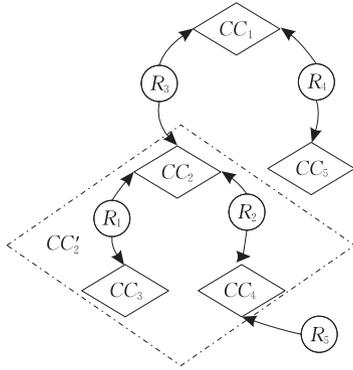


图 3 候选关注点分组示例图

图 3 中菱形表示一个候选关注点, 圆形表示一个需求, 在图中我们可以看出, R_1 依赖于 CC_2, CC_3 , R_2 依赖于 CC_2, CC_4 . 此时通过分析候选关注点和需求本身, 认为 CC_3 和 CC_4 是 CC_2 的下属候选关注点, 从而将其合并组成 CC_2' , 同时将候选关注点 CC_4 与需求 R_5 间的依赖关系重定向为 R_5 与 CC_2' 间的依赖关系, 同时将 R_1, R_2 与 CC_2, CC_3, CC_4 间的关系重定向为与 CC_2' 的关系. 此时, CC_2' 可以继续参与候选关注点分组过程, 并可能与 CC_1, CC_5 组成一个新的候选关注点.

由于候选关注点分组是一个主观性很强的过程, 需要结合实际的分析才能确定是否可以将几个候选关注点组合在一起, 因此, 本节给出了检测候选关注点是否可以被分组的必要条件.

定义 4. 候选关注点分组验证图 $GCC(CC_{set'})$. 设存在由一些独立候选关注点和候选关注点组组成的集合 $CC_{set'}$, 对于 $CC_i, CC_j \in CC_{set'}$, 若存在某需求 R_n 使得 CC_i, CC_j 与 R_n 之间同时存在依赖关系, 则认为 CC_i 与 CC_j 间存在连通边. 以集合 $CC_{set'}$ 中的元素作为点集, 以 $CC_{set'}$ 中存在的连通边作为边集, 存在一个候选关注点分组验证图 GCC .

候选关注点分组策略: 对于一个候选关注点集 $CC_{set'}$, 若候选关注点集 $CC_{set'}$ 可能被重组为一个候选关注点组, 则 $GCC(CC_{set'})$ 必须是一个连通图. 本文依靠验证图 GCC 的连通性来检验这些候选关

注点之间是否存在联系, 如果图 GCC 不具备连通性, 说明其中存在某些候选关注点 CC_i 与其它候选关注点之间不分享需求, 即不存在联系, 不满足分组的必要条件. 在每次进行分组之后, 按照候选关注点分组中需求依赖重定向策略对需求依赖关系进行重定向, 同时再将重组后的 $CC_{set'}$ 集作为一个新的节点加入候选关注点集 CC_{set} , 并在候选关注点集 CC_{set} 中删去 $CC_{set'}$ 中原有的候选关注点. 完成候选关注点分组后, 将 CC_{set} 中包含的候选关注点定义成关注点.

3.2.3 横切关注点验证策略

在本文 3.1 节中提到了横切的关注点的判定方法, 由于横切关注点的判定同样需要人工的分析, 这里给出判定横切关注点的必要条件.

验证横切关注点策略: 存在一个关注点集合 $C_{set'}$, 当且仅当任意属于 $C_{set'}$ 的 C_i 都与某需求 R_j 存在依赖关系, $C_{set'}$ 中才可能存在横切关注点. 若 C_n 为 $C_{set'}$ 中基于 R_j 的横切关注点, 则 $C_{set'}$ 中被 C_n 横切的关注点为实现需求 R_j 必须依靠 C_n 中的功能.

4 基于本体的关注点模型定义及验证方法

基于本体的模型定义方法主要研究的是需求分析所建立关注点模型其本质上有什么样的对象、过程、属性和关系, 以及它们之间的依赖关系如何. 对于一个基于关注点的面向方面模型, 其建模过程更多的是依赖于需求分析人员的经验, 因此, 检验模型的可靠性是本文的研究重点. 第 3 节介绍了从原始需求文本获取关注点以及相关的验证方法, 本节将介绍基于本体的关注点模型, 并给出相关约束来验证关注点模型的完整性和一致性.

4.1 基于本体的关注点模型

关注点模型中包括需求集 R 、关注点集 C 、关注点关联关系集 RL 、横切关注点集 C_{cut} , 在关注点的相关定义中, 可以看出一个关注点是由名字、描述、依赖需求集、类型所组成的四元组, 同时两个关注点之间的关联关系根据关注点类型的不同存在 6 种类型. 一个合理的本体模型必须保证与原始模型描述相同的内容, 根据关注点模型的特点和本体描述的特点, 本文对基于本体的关注点模型有如下定义.

定义 5. 基于本体的关注点模型 CM . 一个基于本体的关注点模型 CM 被定义成一个 4 元组, $CM = (C, RL, RA, Res)$. 其中:

C 代表 CM 中的概念集合. $C = Con \cup Re$, Con 为关注点概念集, Re 为需求概念. $Con = CB \cup CC$, CB 为基础关注点概念集, CC 为横切关注点概念集, 同时 $CB = CtB \cup CkB \cup CiB \cup CpB$, 分别表示 topic、kinds、instance、property 4 种类型的基础关注点, $CC = CkC \cup CiC$, 分别表示 kinds 和 instance 两种类型的横切关注点.

RL 代表 CM 中的关系集合, $RL = Rr \cup Rc$, 其中 Rr 表示 Re 与 Con 之间的依赖关系, $Rc = Rcut \cup$

$Rtype$, 其中 $Rcut$ 代表 CB 与 CC 之间横切与被横切的关系, $Rtype$ 代表 Con 中各种类型的关注点依赖关系.

RA 代表 CM 中概念间的层次关系集合.

Res 代表 CM 中的约束集合.

根据定义 5 中所定义的本体关联模型 CM, 本文构建的其概念间的关联如图 4 所示.

表 1 和表 2 中给出了基于本体的关注点模型中所定义的概念及关系.

表 1 基于本体的关注点模型中的关系

关系	形式	含义
relay	requirement->concern	需求实例通过 relay 关系关联到关注点上
dependent	concern->concern	没有定义域值域限制的的关注点依赖关系
motivate	concern->concern	dependent 的反向关系
crosscutting	crosscutting_concern->concern	横切关注点与被其横切的关注点之间的关系
be_crosscut	concern->crosscutting_concern	被横切关注点与其横切关注点关系 crosscutting 的反向关系
kinds_of	kindsC/kindsB->kindsC/kindsB	左侧本体关系为表示不同类型关注点之间的不同关联关系
instance_of	kindsC/kindsB->instanceC/instanceB	例如: relate_to 表示一个 kinds 类型关注点
relate_to	topic->kindsB/kindsC	可以通过一个 relate 类型的关联关系关联到一个 kinds 类型横切关注点.
applies_of	instanceC/instanceB->property	

表 2 基于本体的关注点模型中的概念

概念	含义
requirement	需求
concern	关注点
crosscutting_concern	横切关注点
base_concern	非横切关注点
instanceC	instance 类型横切关注点
kindsC	kinds 类型横切关注点
topic	topic 类型非横切关注点
instanceB	instance 类型非横切关注点
kindsB	kinds 类型非横切关注点
property	property 类型非横切关注点

本文通过总结归纳现有基于本体的需求模型完整性和一致性的约束规则, 结合面向方面需求模型的实际特点, 定义和实现了 CM 中的约束集 Res . 约束集 $Res = Rc \cup Ri$, 其中 Rc 为完整性约束集, Ri 为一致性约束集, 具体的约束如下所示:

完整性约束集 Rc

$Res.1$ 领域可扩展性约束: 一般性的软件需求都应该包括软件安全需求、功能性需求和非功能性需求. 本约束规则在此基础上通过对不同软件产品的本体知识模型进行分析, 定义关注点模型中所必须要定义的关注点, 如:

- (1) 必须存在安全需求关注点.
- (2) 必须存在非功能性需求关注点.

$Res.2$ Topic 约束: 在本体实例中, 有且仅有一个 topic 概念的实例.

$Res.3$ 需求覆盖约束: 对于任何一个 requirement

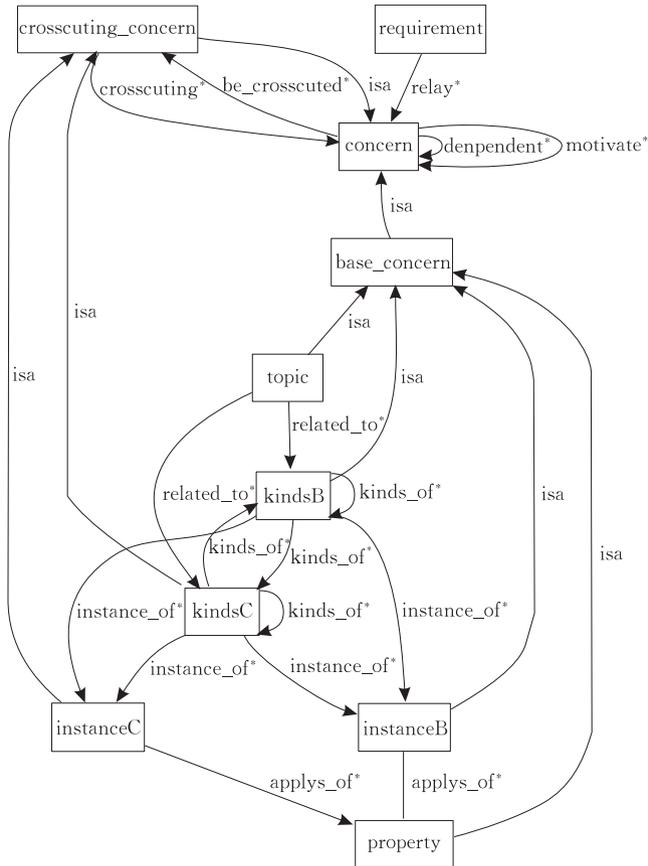


图 4 基于本体的关注点概念间的关联

的实例, 都最少依赖于一个关注点.

$$requirement \sqsubseteq \geq 1 \text{ relay.concern.}$$

$Res.4$ 关注点需求依赖约束: 每个关注点必定

被 1 个以上的需求所依赖.

$\text{concern} \sqsubseteq \geq 1 \text{ relay}^- . \text{requirement}.$

一致性约束集 R_i

Res.5 实例命名互斥约束: 每一个概念具有不同名称的实例.

Res.6 关系集约束: 任何一个合法关系 RL 必须建立在两个合法的概念 C 上.

$\forall RL. \top \sqsubseteq C; \forall RL^- . \top \sqsubseteq C.$

Res.7 关系定义域、值域约束: 为 *Res.6* 的扩展定义, 任何一个合法的关系 RL , 必须满足表 2 中所定义的定义域和值域. 例如:

$\forall \text{kinds_of}. \top \sqsubseteq \text{kindsB} \sqcup \text{kindsC},$

$\forall \text{kinds_of}^- . \top \sqsubseteq \text{kindsB} \sqcup \text{kindsC}.$

Res.8 横切约束: 对于任意一个 $\text{crosscutting_concern}$ 的实例, 最少横切两个关注点.

$\text{crosscutting_concern} \sqsubseteq \geq 2 \text{ crosscutting.concern}.$

Res.9 互逆关系约束: 关系 crosscutting 与 be_crosscut ; motivate 与 dependent 具有互逆性.

$\text{crosscutting} \equiv \text{be_crosscut}^- ,$

$\text{motivate} \equiv \text{dependent}^- .$

Res.10 关注点关联关系存在约束: 当且仅当两个关注点包含的需求集存在交集, 这两个关注点才存在关联关系.

对于上述定义的 10 条约束中的 *Res.3*、*Res.4*、*Res.6*、*Res.7*、*Res.8*、*Res.9*, 文中给出了基于描述逻辑 (Description Logic) 的定义, 直接作为公理存储在 本体当中, 用于推理机进行逻辑检测. 对于 *Res.1*、*Res.2*、*Res.5*、*Res.10* 本文利用 Jena API 工具通过对 本体进行查询和推理来验证约束条件是否成立. 例如检验 *Res.10* 的 Jena API 算法如下所示:

Jena API: 查询所有关注点 C 的依赖需求集 R_{set} , 当 C_i 与 C_j 间定义了关联关系时, 若 $R_{\text{set}i}$ 与 $R_{\text{set}j}$ 的交集不为空, 则说明 C_i 与 C_j 间关联关系成立, 否则说明不成立. 若所有关注点关联关系都成立, 则返回 true, 否则返回 false.

4.2 关注点模型到基于本体的关注点模型的映射及验证

本节介绍了从关注点模型到基于本体的关注点模型的映射规则及映射后的验证方法, 设存在一个需求集 R_{set} 、一个关注点集合 $C_{\text{set}}(id, Des, Rs, type)$ 、一个关注点关联关系集合 $RL_{\text{set}}(C_i, C_j, reqLs, type, crosscutting)$ 和一个横切关注点集合 $Ccut_{\text{set}}$, 有如下映射规则:

(1) 若有 $C_i \in Ccut_{\text{set}}$ 且 $C_i \in C_{\text{set}}$, 如果 $C_i.type =$

kinds , 则建立名为 $C_i.id$ 的 kindsC 概念实例, 如果 $C_i.type = \text{instance}$, 则建立名为 $C_i.id$ 的 instanceC 概念实例.

(2) 若有 $C_i \notin Ccut_{\text{set}}$ 且 $C_i \in C_{\text{set}}$, 如果 $C_i.type = \text{topic}$, 则建立名为 $C_i.id$ 的 topic 概念实例, 如果 $C_i.type = \text{kinds}$, 则建立名为 $C_i.id$ 的 kindsB 概念实例, 如果 $C_i.type = \text{instance}$, 则建立名为 $C_i.id$ 的 instanceB 概念实例, 如果 $C_i.type = \text{property}$, 则建立名为 $C_i.id$ 的 property 概念实例.

(3) 对于 $\forall R_i \in R_{\text{set}}$, 建立名为 R_i 的 requirement 概念实例.

(4) 对于 $\forall C_i \in C_{\text{set}}$ 中 $\forall R_i \in C_i.Rs$, 建立关系 $\text{relay}(R_i, C_i)$.

(5) 对于 $\forall RL_i \in RL_{\text{set}}$,

若 $RL_i.\text{crosscutting} = 1$, 则建立关系

$\text{crosscutting}(RL_i.C_i, RL_i.C_j)$ 、

$\text{be_crosscut}(RL_i.C_j, RL_i.C_i)$;

若 $RL_i.type = \text{kinds_of}$, 则建立关系

$\text{kinds_of}(RL_i.C_i, RL_i.C_j)$;

若 $RL_i.type = \text{instance_of}$, 则建立关系

$\text{instance_of}(RL_i.C_i, RL_i.C_j)$;

若 $RL_i.type = \text{relate_to}$, 则建立关系

$\text{relate_to}(RL_i.C_i, RL_i.C_j)$;

若 $RL_i.type = \text{applies_of}$, 则建立关系

$\text{applies_of}(RL_i.C_i, RL_i.C_j)$;

若 $RL_i.type = \text{motivate}$, 则建立关系

$\text{motivate}(RL_i.C_i, RL_i.C_j)$;

若 $RL_i.type = \text{dependent}$, 则建立关系

$\text{dependent}(RL_i.C_i, RL_i.C_j)$.

通过上述的映射规则, 我们可以在关注点模型的基础上获得相应的基于本体的关注点实例 $CM_instance$, 之后通过 CM 中的约束来检验 $CM_instance$ 的完整性和一致性.

定义 6. 验证函数 $Verify$.

$Verify(CM_instance, Res) =$

$\begin{cases} \text{true}, & \text{if the } CM_instance \text{ 满足 } Res, \text{ 否则} \\ \text{false}, & \text{else} \end{cases}$

本文利用描述逻辑和 Jena API 实现了检验各项约束的工具, 此时利用定义的验证函数 $Verify$ 来检验 $CM_instance$ 的完整性和一致性. 若函数返回结果为真, 代表通过检验, 否则需要对关注点模型进行相关修改.

5 实例分析

本节将针对一个实例,检验基于本体的关注点模型的完整性和一致性验证策略,并给出分析结果.文献[15]中给出的是一个电灯控制系统的需求文本,该需求文本描述规范,被多次作为需求分析实例所使用,包括本文中介绍到的面向方面稳态驱动需求方法、SCR(Software Cost Reduction)方法等.需求文本描述了电灯控制系统的各方面具体需求.在功能性需求方面,分别从用户需求角度($U\langle number \rangle$)和管理员需求角度($FM\langle number \rangle$)给出了系统的

功能需求,需求文本又从容错度、安全性、用户界面这 3 个方面描述了系统的非功能性需求($NF\langle number \rangle$).

实验人员依据第 3 节中介绍的分析方法和建模过程,首先建立需求集 R ,并对 R 进行词法分析、需求重审,获得改进后的需求集 R 和候选关注点集 CC_{set} ,之后通过自顶向下和自底向上的分析方法并结合候选关注点分组策略获得关注点集和关注点关联关系集,最后通过横切判定并结合横切关注点验证策略获得横切关注点集.定义了一个未经验证的关注点模型,其中关注点定义和关注点关联图如表 3 和图 5 所示.

表 3 未经检测的电灯控制系统中的关注点定义

关注点名	描述	依赖需求集	类型
SYS	The system		Topic
LS	Lights system	{FM1,FM2,FM4,FM6,FM7,NF3~NF5,U13,U14}	Topic
FM	Facility manager	{FM4,FM6,FM7,NF5}	Kinds
USER	User related function	{FM2,NF3,U14}	Kinds
CS	Control system	{FM1,FM2,NF3~NF5,U13,U14}	Kinds
FMCP	Facility manager control panel	{FM4,FM6,FM7}	Kinds
SIG	Signal	{FM1,U14}	Kinds
JO	Judge occupied	{FM2,FM6,NF3,NF4,U13,U14}	Instance
JC	Judge control	{NF5}	Instance
MDMAL	mdmalfunction	{FM7,NF4}	Instance
OLSMAL	olsmalfunction	{FM7,NF3}	Instance
T2i	Manager controls timing	{FM4}	Instance
MD	Motion detector	{U14}	Instance
OLS	Outdoor light sensor	{FM1}	Instance
SI	Safe illumination	{U13}	Instance
TIME	Timing	{FM2,FM4}	Property
Override	Override control	{FM6,FM7}	Instance
LC	Light control	{FM1,FM2,NF3,NF5,U14}	Instance
MAL	Malfunction	{FM7,NF3,NF4}	Kinds

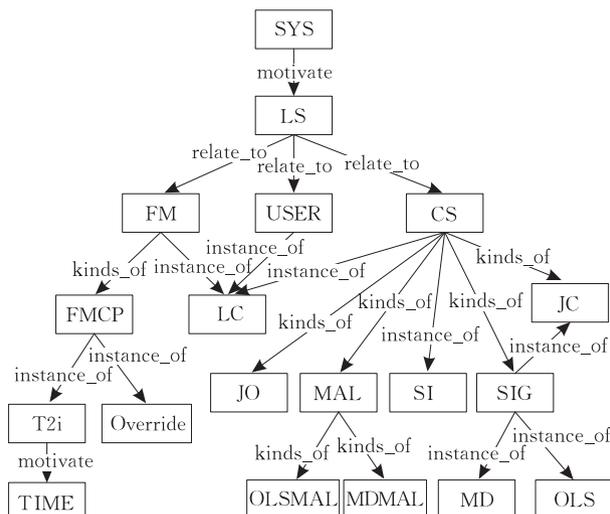


图 5 未经检测的电灯控制系统的关注点关联图

这部分关注点模型的需求集 R 中包含 {FM1, FM2, FM4, FM6, FM7, NF3~NF9, U13, U14}, 横

切关注点集 C_{cut} 中包含 {FMCP, MAL, SIG}. 此时,依据 4.1 节中定义的基于本体的关注点模型 CM 和 4.2 节中给出的关注点模型到 CM 的映射规则,将关注点模型映射到 CM 之上获得 $CM_{instance}$,图 6 中给出了 $CM_{instance}$ 中的一部分.包括需求 NF1,FM2,FM4,关注点 FMCP、Override、T2i 以及它们之间的关系.

再利用定义 6 中给出的验证函数,通过进行本体推理机推理,检测该模型的完整性和一致性.在 Intel Core Duo E7500 CPU、主频 2.93GHz、2GB 内存的运行环境下,检测程序运行时间共 862ms,检测出 20 余项错误,其中包括的主要错误有:

(1) 未定义基于安全和非功能性需求的关注点.

(2) 关注点间关系定义错误,例如 MAL 到 OLSMAL 和 MDMAL 的关系不能为 kinds_of 而应该是 instance_of.

(3) 需求未获得覆盖,如 NF8, NF9 等需求不存

在与任何关注点间的依赖关系。

(4) SYS 关注点并未被任何需求所依赖。

(5) 一些本来不应该存在关联关系的关注点间被定义了关联关系,如 SIG 和 JC 关注点。

(6) 存在两个以上的 Topic 类型关注点。

存在上述错误的主要原因包括:需求建模实

验人员对领域知识掌握不够充分、建模过程中对细节的定义存在纰漏、对需求文本理解的不够深刻等等。

根据工具检测报告,我们对原有的关注点模型进行了修改,定义修改后的关注点和关注点关联图如表 4 和图 7 所示。

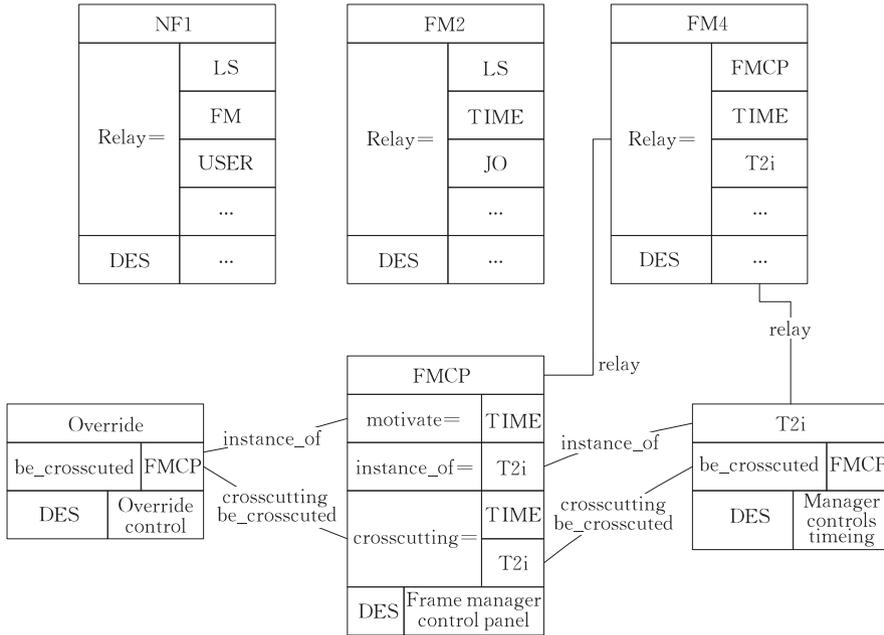


图 6 利用映射规则构建的部分本体实例图

表 4 修改后的电灯控制系统中的关注点定义

关注点名	描述	依赖需求集	类型
SYS	The system	{FM1, FM2, FM4, FM6, FM7, NF3~NF9, U13, U14}	Topic
SEC	Security	{NF6, NF7}	Kinds
UI	User Interface	{NF8, NF9}	Kinds
ADS	According to DIN standards	{NF6}	Kinds
NHC	No hazardous conditions	{NF7}	Kinds
EIU	Easy and intuitive to use	{NF8}	Kinds
WUI	Warning on unreasonable inputs	{NF9}	Kinds
LS	Lights system	{FM1, FM2, FM4, FM6, FM7, NF3~NF5, U13, U14}	Kinds
FM	Facility manager	{FM4, FM6, FM7, NF5}	Kinds
USER	User related function	{FM2, NF3, U14}	Kinds
CS	Control system	{FM1, FM2, NF3~NF5, U13, U14}	Kinds
FMCP	Facility manager control panel	{FM4, FM6, FM7}	Kinds
SIG	Signal	{FM1, U14}	Kinds
JO	Judge occupied	{FM2, FM6, NF3, NF4, U13, U14}	Instance
JC	Judge control	{NF5}	Instance
MDMAL	mdmalfunction	{FM7, NF4}	Instance
OLSMAL	olsmalfunction	{FM7, NF3}	Instance
T2i	Manager controls timing	{FM4}	Instance
MD	Motion detector	{U14}	Instance
OLS	Outdoor light sensor	{FM1}	Instance
SI	Safe illumination	{U13}	Instance
TIME	Timing	{FM2, FM4}	Property
Override	Override control	{FM6, FM7}	Instance
LC	Light control	{FM1, FM2, NF3, NF5, U14}	Instance
MAL	Malfunction	{FM7, NF3, NF4}	Kinds

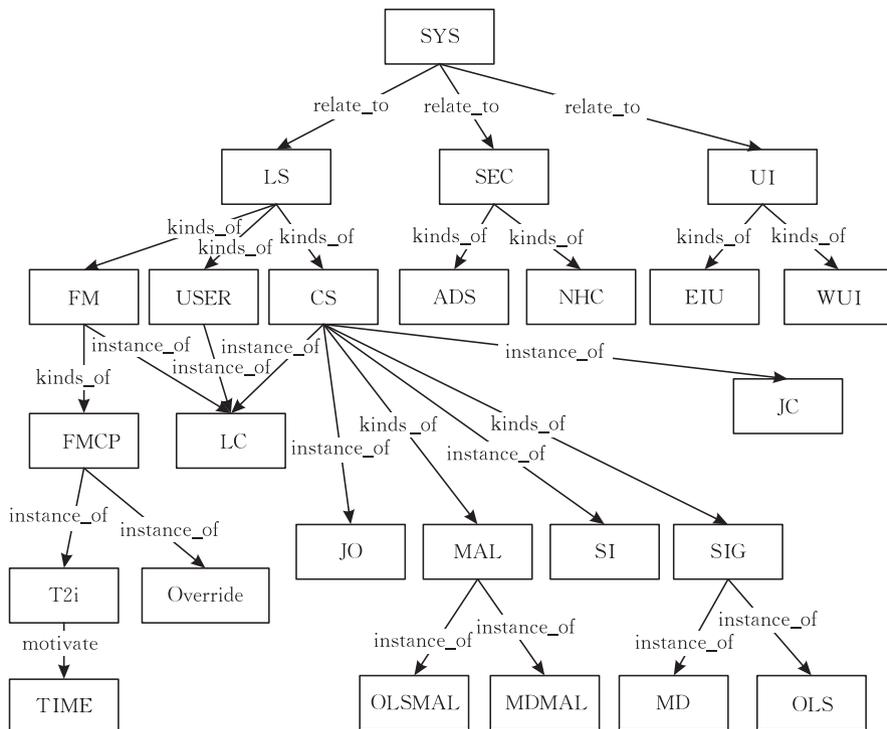


图7 修改后的电灯控制系统的关注点关联图

经过检测,修改后的关注点模型满足文中定义的全部约束,并依据对实际的需求文本进行分析,检验了该模型的完整性和一致性,验证了本文提出方法的有效性。

6 相关工作比较

需求模型的验证是当前软件工程领域的热点问题之一,很多研究人员作了广泛而深入的研究工作。本文将从面向方面需求验证和基于本体的需求验证两个方面阐述相关工作。

面向方面的需求方法是被认为在模块化需求、确定和解决需求冲突方面具有一定的优势,关于面向方面的需求检测方面已经有一些较好的方法。Araujo 等人在文献[6]中提出了一种利用状态机分别构建横切关注点和非横切关注点模型,并基于状态机结合的方法检验需求冲突。Brito 等人在文献[7]中提出了一种多重标准决策定义方法,该方法中对非功能化关注点依据其对于开发的重要程度进行加权,通过分析非功能关注点及其权值的关系,从一个更高的角度检测需求间的冲突。Mehner 等人在文献[8]中定义了面向方面需求和 UML 模型间的关联联系,通过分析 UML 模型来检验需求模型中的完整性和一致性。Lancaster 大学的 Weston 在文献[9]中提出了一种形式化的面向方面需求语义冲突检查,

采用 RDL (Requirement Description Language) 描述需求,并依靠对需求的分类化的语义分析检测需求中的冲突。Sampaio 等人在文献[16]中给出了 EA-Miner 工具,采用文本检测的方法从需求文本中获取关注点,并可以自动生成基于面向方面需求方法的关注点模型,同时该论文中还提到了一些应用的方法来保证建模的准确性。Sardinha 等人在文献[17]中提出了一个面向大规模需求文本的自动冲突检测工具 EA-Analyzer,基于贝叶斯学习方法,对描述需求文本的 RDL 进行需求冲突检测。这些方法的侧重点主要是利用 RDL 自然描述语言,并通过语义分析来进行的需求验证。上述方法基本上都是通过一定的方法和技术对需求文本中存在的冲突进行检测,而缺少对构建需求模型验证的方法。

由于本体具有严谨的定义方法和强大的约束、推理能力,因此基于本体的需求分析及验证方法研究也有很多。2000 年,陆汝钤等人在文献[18]中提出了一个面向本体的需求分析方法,文中用本体作为对象关联的手段,以增强面向对象方法的表达能力。Liu 在文献[10]中定义了一套基于本体的需求冲突分析方法,该方法基于需求的动作图给出本体模型的定义,并建立相应的冲突检测规则来分析需求中存在的冲突。陈等人在文献[1]中提出了一个采用本体模型的基于问题框架的需求建模,对于问题框架需求分析方法定义了一个有效的建模工具,同

时定义了多条完整性和一致性规则来验证所建立本体模型的准确性和一致性. Guo 等人在文献[11]中提出了一个基于本体元模型和一阶逻辑检测需求特征模型一致性的检验方法. 文章针对特征模型的特点, 假定已有特征模型满足一致性, 用一个特定的冲突检测集合来验证对特征模型进行删除、修改等操作后模型是否依然满足一致性. Lee 等人在文献[19]中提出了一个基于本体的需求工程框架, 文章通过分析多种需求框架的特点, 将它们整合在一个本体模型之上, 为软件集成系统提供了一个良好的框架. Siegemund 等人在文献[20]中提出了一个面向目标需求分析的本体模型, 依据面向目标模型定义本体模型, 以此来检验面向目标模型的准确性和一致性. 然而利用本体对需求直接进行建模的方法存在着很多的不完善性, 对于不同需求方法所构建的基于本体的需求模型验证又有很大的局限性, 即仅适用于所针对的需求方法, 目前缺少针对面向方面需求方法采用基于本体来进行需求模型验证的相关工作.

本文针对面向方面需求分析方法, 提出了在构建需求模型的过程中所应该遵守的策略, 同时构建了基于本体的关注点模型, 并在该模型中定义了一系列约束, 利用作者开发的验证工具可以有效地自动化检验需求模型的完整性和一致性, 为需求模型的形式化验证提供了基础.

7 总结和展望

本文给出了一种基于面向方面需求分析模型的验证方法, 在从原始需求文本生成关注点模型的过程中给出一定的验证策略, 同时文中构建了基于本体描述的关注点模型 CM, 并给出了从关注点模型到模型 CM 的映射规则, 通过定义 CM 中存在的约束并利用描述逻辑和 Jean API 实现约束检验工具, 验证了关注点模型的完整性和一致性, 为设计和实现一个可靠的软件系统提供了有效的信息和重要的支持.

今后的主要工作是应用其它实例以进一步检验本文所提方法的有效性和实用性, 并给出完整的工具支持, 同时针对面向方面文档驱动需求分析方法框架, 给出针对形式化需求文档的一致性、完整性验证策略.

参 考 文 献

- [1] Chen Xiao-Hong, Yin Bin, Jin Zhi. Ontology-guided requirements modeling based on problem frames approach. *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. *软件学报*, 2011, 22(2): 177-194)
- [2] van Lamsweerde A. Goal-oriented requirements engineering: A guided tour//*Proceedings of the 5th IEEE International Symposium on Requirements Engineering*. Washington, USA, 2001: 249-261
- [3] Rashid A. Early aspects: A model for aspect-oriented requirements engineering//*Proceedings of the 10th IEEE International Requirements Engineering Conference (RE'02)*. Washington, USA, 2002: 199-202
- [4] Batory Don S. Feature models, grammars, and propositional formulas//*Proceedings of the 9th International Conference on Software Product Lines (SPLC'05)*. Rennes, France, 2005: 7-20
- [5] Zhang Jia-Zhong, Wang Zhi-Jian, Yi Bo, Xu Jia-Fu. A hierarchical object-oriented software requirements model and its mechanical support. *Journal of Software*, 1998, 9(6): 414-418(in Chinese)
(张家重, 王志坚, 尹波, 许家福. 对象式软件需求模型及其机器支撑. *软件学报*, 1998, 9(6): 414-418)
- [6] Araujo Joao, Whittle Jon, Kim Dae-Kyoo. Modeling and composing scenario-based requirements with aspects//*Proceedings of the 12th IEEE International Requirements Engineering Conference (RE'04)*. Washington: IEEE, 2004: 122-131
- [7] Brito Isabel Sofia, Vieira Filipe, Moreira Ana, Ribeiro Rita A. Handling conflicts in aspectual requirements compositions. *Transactions on Aspect Oriented Software Development*, 2007, 4620(2007): 144-146
- [8] Mehner Katharina, Monga Mattia, Taentzer Gabriele. Interaction analysis in aspect-oriented models//*Proceedings of the 14th IEEE International Requirements Engineering Conference (RE'06)*. Washington, USA, 2006: 66-75
- [9] Weston Nathan, Chitchyan Ruzanna, Rashid Awais. Formal semantic conflict detection in aspect-oriented requirements. *Requirements Engineering*, 2009, 14(4): 247-268
- [10] Liu Chi-Lun. Ontology-based requirements conflicts analysis in activity diagrams//*Proceedings of the International Conference on Computational Science and Its Applications*. Berlin, Germany, 2009
- [11] Guo Jianmei, Wang Yinglin, Trinidad Pablo, Benavides David. Consistency maintenance for evolving feature models. *Expert Systems with Applications*, 2012, 39(5): 4987-4998
- [12] Jin Ying, Hao Weiping, Ma Pengfei. Concern based approach to generating SCR requirement specification: A case study//*Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering (CSIE 2009)*. Washington, USA, 2009: 695-700
- [13] Hamaza Haitham S. Separation of concerns for evolving systems: A stability-driven approach//*Proceedings of the Workshop on Modeling and Analysis of Concerns in Software (MACS 2005)*. New York, USA, 2005: 1-5
- [14] Jin Ying, Zhang Jing, Hao Weiping, Ma Pengfei, Zhang Yan, Zhao Haiyan, Mei Hong. A concern-based approach to

generating formal requirements. *Frontiers of Computer Science in China*, 2010, 4(2): 162-172

- [15] Queins Stefan, Zimmermann Gerhard, Becker Martin, Kronenburg Martin, Peper Christian, Merz Rolf, Schaefer Juergen. The light control case study: Problem description. *Journal of Universal Computer Science*, 2000, 6(7): 586-596
- [16] Sampaio A, Chitchyan R, Rashid A, Rayson P. EA-Miner: A tool for automating aspect-oriented requirements identification//*Proceedings of the 20th IEEE ACM International Conference on Automated Software Engineering (ASE 05)*. New York, USA, 2005: 352-355
- [17] Sardinha Alberto, Chitchyan Ruzanna, Weston Nathan, Greenwood Phil, Rashid Awais. EA-analyzer: Automating conflict detection in a large set of textual aspect-oriented

requirements. *Automated Software Engineering*, 2012

- [18] Lu Ru-Qian, Jin Zhi. Ontology-oriented requirements analysis. *Journal of Software*, 2000, 11(8): 1009-1017 (in Chinese)
(陆汝钊, 金芝. 面向本体的需求分析. *软件学报*, 2000, 11(8): 1009-1017)
- [19] Lee Seok Won, Gandhi Robin A. Ontology-based active requirements engineering framework//*Proceedings of the 12th Asia-Pacific Software Engineering Conference*. Washington, USA, 2005: 481-490
- [20] Siegemund Katja, Thomas Edward J, Zhao Yuting, Pan Jeff, Assmann Uwe. Towards ontology-driven requirements engineering//*Proceedings of the 10th International Semantic Web Conference*. Koblenz, Germany, 2011: 1-15



JIN Ying, born in 1971, professor, Ph. D. supervisor. Her main research interests include software requirements engineering, formal methods, programming language and its implementation.

LIU Hua-Xiao, born in 1986, Ph. D. candidate. His main research interests include software requirements engineering, formal methods of software development.

ZHANG Peng, born in 1986, Ph. D. candidate. His main research interests include semantic web and ontology engineering.

Background

The verification of requirements analysis model is one of the main issue of requirements engineering, which is playing an important role in software quality assurance. A lot of researches on requirements modeling and requirements model verification have been conducted and achieved good results in certain aspects. Most existing approaches to verifying Aspect-Oriented requirements model focus on composition of crosscutting concerns and more attention has been paid in semantic analysis.

In this paper, under the framework of aspect-oriented document driven requirements engineering an approach to verifying requirements model has been proposed. At first the

validation

strategy during the requirements modeling process has been given, then an ontology based concerns model and its verification has been introduced. A case study has been done and the effectiveness of our approach has been indicated.

This work was supported in part by the New Century Excellent Talents in University (NCET-10-0436). The main research of this project is to develop methods, techniques and tools for assuring high confidence of software during requirements, design, testing and execution. The work of this paper is related to verification of requirements model, which is one of the project's subtasks.