

一个设计模式自动识别技术研究框架

周晓宇^{1,2} 徐宝文^{1,2,3}

(东南大学计算机科学与工程学院 南京 211189)¹ (江苏省软件质量研究所 南京 211189)²
(武汉大学软件工程国家重点实验室 武汉 430072)³

摘要 在程序代码中自动识别设计模式可以为面向设计模式的软件理解、维护和重构等活动提供自动化支持。比较系统地分析了该领域十余年来研究状况和存在的问题,提出了一套以准确和通用的设计模式识别技术为目标,以层次化识别特征的确定、基于静态分析技术的语义特征发掘与识别技术、基于语义特征和排除性约束的软件框架描述技术、基于描述逻辑的设计模式形式化技术等为内容的研究框架。介绍了各项技术的基本思想和主要内容,论述了其可行性。

关键词 设计模式,软件分析,逆向工程

中图法分类号 TP311 **文献标识码** A

New Research Framework for Automatic Recovery of Design Pattern

ZHOU Xiao-yu^{1,2} XU Bao-wen^{1,2,3}

(School of Computer Science & Engineering, Southeast University, Nanjing 211189, China)¹

(Jiangsu Institute of Software Quality, Nanjing 211189, China)²

(State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China)³

Abstract Automatic recovery of design patterns in codes can promote automation of program comprehension, maintenance and refactoring. Researches on design pattern recovery during the past decade and more were summarized, and shortcomings of existing techniques were pointed out. To solve the problems and achieve more accurate and general recovery technique, a research framework was introduced that is made up of researches on multilevel recovery target, precise definition and identification of semantic features based on static program analysis, software framework description based on semantic features and exclusive rules, formal design pattern description technology based on description logic. The main idea was presented and feasibility of the framework was demonstrated.

Keywords Design pattern, Program analysis, Reverse engineering

1 引言

设计模式是软件设计中典型问题的典型解决方案,重在提高软件的可重用性和可扩展性。应用设计模式设计和实现软件是将设计模式映射到程序代码的过程,而设计模式的识别旨在建立从程序代码到设计模式的映射。设计模式识别研究对于软件理解与维护具有重要的意义:从遗产软件中识别出设计模式可以实现设计层面的软件理解,保持设计概念的可追踪性;也只有识别出源程序中的设计模式,才可以在维护中利用现有模式的可重用和可扩展潜力,反之则可能破坏已有模式;很多遗产软件还没有使用设计模式,如果能够从中识别出可重构为设计模式的代码,就可以有效地实现软件的重构、升级。

设计模式识别研究的另一个重要作用是可以使得现有的设计概念得到明确的、可识别的定义。现有设计模式的主流

描述多采用图示、自然语言、样例相结合的方式,其中大量概念未经明确定义。目前的正向工程实践对此尚能接受,而设计模式的自动识别研究的第一步就是要给出严格定义的识别特征。这种定义必须在不损害灵活性的前提下使模式的本质属性和灵活实现空间都得到明确、具体的界定。这种严格定义是设计模式应用工具化、自动化的前提。

设计模式的识别在本质上是一种面向对象软件中对象交互模式的识别,此项研究可以深入探讨描述和识别对象交互所需要的各项技术,其成果对软件体系结构的识别也具有比较好的借鉴意义。本文比较系统地分析和总结了现有设计模式自动识别技术中存在的问题,针对这些问题提出了一个系统、可行的研究框架。

2 现状与分析

2.1 识别目标的确定

收稿日期:2008-06-25 本文受国家杰出青年科学基金项目(60425206),国家自然科学基金资助项目(60503020),江苏省自然科学基金(BK2006094)资助。

周晓宇(1972-),男,副教授,主要研究方向为程序设计语言、软件分析与逆向工程、软件测试等;徐宝文(1961-),男,教授,博士生导师,主要研究方向为程序设计语言、软件工程、并行与网络软件等。

Erich Gamma 等对设计模式的描述虽然是公认的参照基准,但他们的工作以正向工程为目的,对应用模式后的程序特征仅给出了一些示例和一般性的讨论,没有明确界定模式的核心与非核心成分、根本属性与非根本属性的界限^[1]。由于设计模式涉及的层面较多,现有的设计模式识别研究中,不同文献对识别模式成分和特征的选择、描述差异较大,导致研究成果(包括实验数据)难以比较,说服力不足。例如,文献[3, 7, 10, 16, 33, 34]对“观察者”模式的成分、特征的选择和描述差异很大,识别效果难以比较。

2.2 模式特征识别研究

设计模式要求通过特定的结构形态实现特定的语义,其特征可以从语法和语义两方面来描述和识别。目前对其结构形态即语法特征的研究很多,也比较成熟^[3, 11-15, 18-23]。但仅有语法是远远不够的。有些设计模式在静态结构上十分接近甚至完全相同。例如,Ilka Philippow 为策略和状态两个模式给出了完全相同的语法特征^[3]。有些模式虽然具有鲜明的语法特征,但符合这些特征的代码也未必就能实现相应的语义。例如,单件模式的语法结构很独特(Ilka Philippow 等进行了比较明确的限定^[3]),但一段代码即使满足这些特征,仍然可以创建出多个共存的个体,违反“单件”的要求。

有些研究在无法准确定义语义特征的情况下用一些语法约束替代语义特征。设计模式中的一些设计概念仅对语义效果有要求,对实现方式并无限定。使用语法约束替代语义特征不能适应复杂多变的实现方式,缺少灵活性。对很多语义特征也很难找到合适的语法特征来替代。例如,观察者模式中的“登记”行为要求观察者和被观察者间动态地建立关联。由于没有找到合适的方法来定义这种关联,有人将以“Add”等字样开头或把观察者类型的参数放在赋值号右端的方法认作登记行为^[4, 16]。这样的特征描述不能反映模式的本质,自然也不能得到准确的结果。

还有少量研究提出使用公理语义描述角色行为的动态语义^[10, 16, 33, 34, 38],然后在程序执行过程中进行动态监测、验证思想。此类研究有两个局限:(1)每种语义描述都有其优势和局限,有些模式行为并不强调操作前后的状态变换,例如适配模式中的适配行为;(2)需要检测的公理语义的动态分析通常不是角色数据(在设计模式中承担职责的类数据成员)的变化,而是它们之间语义关联的变化,角色数据因应用而异,现有研究(包括动态语义分析本身)尚未提供识别角色数据和角色数据语义关联的能力;(3)动态监测必需明确监测的具体对象,并将监测代码插入源程序的适当位置,需要建立在对源程序充分理解的基础上,因此对软件理解意义不大。

因此,欠缺定义和识别模式语义特征的能力是目前设计模式识别研究的主要缺陷,导致识别准确性不高,灵活性不足。

2.3 模式范围的界定

现有研究中,若识别出某个类属于某个模式,则认为这个类的所有数据和方法都属于这个模式。这种识别是比较粗糙的。在实际软件中,即便一个类的部分方法和数据成员在模式中承担职责,其它成员也可能承担基于应用的其他职责,甚至一个类的不同成分分属不同的模式。因此,要界定设计模式的范围必须识别所有在模式中承担职责的方法和数据,即角色方法(行为)和角色数据。目前的设计模式识别研究仅针

对少量的角色行为,尚未见识别角色数据的研究。

2.4 形式化描述技术

要识别设计模式,必须有一套严格、无歧义的形式化体系描述识别对象的成分与属性。在目前的识别研究中,一个重要障碍就是缺乏适当的设计模式形式化描述技术。

从描述内容来讲,现有研究存在以下问题:(1)主要集中在语法特征描述方面,对行为描述的研究很少;(2)语义特征贫乏,缺少严格定义和系统描述。

从采用的形式化手段来讲,主要存在以下问题:(1)大多数研究采用朴素的关系代数描述成分间的关联,这种方式虽然方便、灵活,但不成体系,推演能力弱,不能代表今后的发展方向;(2)设计模式描述涉及的层面较多,包括类体系结构特征、方法接口特征、行为序列、行为效果、行为语义、行为间或行为与数据间的语义关联等,目前的形式化描述体系只能有效地描述其中的某一方面,不够成一个完整的系统;(3)现有的形式化描述体系不适用于描述设计模式中的行为,具体表现在以下几个方面:

- 基于命题逻辑和集合论的面向对象规格说明技术,如 Object-Z^[30]、对象约束语言(OCL)^[34]、契约(Contract)^[31, 38]等重在描述单个行为的效果,缺乏描述事件序列的能力。

- 状态机技术的缺点在于缺乏有力的证明系统。

- 时态逻辑及其变体^[5, 6]及进程代数类技术^[7-9]具备较强的推演、证明能力,但此两者分别侧重行为的效果和实施。设计模式中既包括对行为效果的要求,也包括对行为实施的要求。现有研究没有将二者进行结合。

- 现有行为描述技术主要是为并发实时系统的描述和验证而设计的,而 23 种 GoF 模式均不涉及并发和实时问题。

- 现有的描述技术是为了完整而准确地描述软件行为而设计的,描述行为框架的能力比较弱。

- 设计模式中行为的主体是类(或对象),并发实时系统中行为的主体是进程,现有技术中描述进程结构的方法不适合描述类结构。

设计模式描述涉及层面较多,概念复杂,设计模式的发展要求描述技术具备可扩展性,这些都说明设计模式的描述具备明显的知识描述的特征。当前已经出现采用基于描述逻辑的本体语言 OWL 描述设计模式的萌芽^[28],但该研究仅关注静态结构方面,它采用 OWL 支持的复杂概念构造子来描述设计模式中的复杂概念以及概念层次,未考虑动态行为间的关联。目前还未见将静态结构和动态行为相结合,使用描述逻辑描述设计模式的研究。

3 一个新的研究框架

当前设计模式识别主要存在两方面的问题,即不够准确、完整、细致以及缺乏通用性和灵活性。造成这种现状的原因

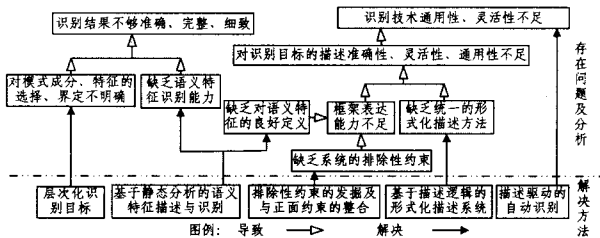


图1 设计模式识别研究存在的问题和相应的解决方法

有多个方面,比如对识别目标缺乏明确的界定、缺少语义描述和识别能力,缺少框架描述能力,缺少综合各层面信息的统一的形式化描述等。我们用图 1 系统地表达了所存在的问题和相应的解决方案。

3.1 层次化的识别目标

为明确设计模式描述和识别所涉及的研究对象和范围,我们认为,可以从 4 个层次来描述设计模式。这 4 个层次是:核心层、典型层退化形态、可重构形态。核心层指必不可少的成份和属性;典型层,指并非必不可少,但在典型实现中经常出现的成份和属性;退化形态指其结构和设计模式之间没有明显的矛盾,但是相对简单,可以扩充为设计模式的代码形态;可重构形态指不具备设计模式所要求的基本结构,但所实现的语义与特定模式一致或相近,因而可以重构为设计模式的代码形态。

目前关于设计模式的非形式化描述中已经涵盖了设计模式中的主要成份和部分约束条件,需要进行的工作主要是进行核心成份和典型成份的划分以及退化形态、可重构形态特征的发掘。层次划分的正确性可以分别通过模式实例以及与正向工程研究的比对进行验证。可重构代码中的数据和行为的语义关联必须符合特定模式中角色数据与角色行为之间的语义关联。

我们认为,这种区分本质和非本质特征的层次划分对明确设计模式概念的内涵很重要。同时,这种层级目标可以为设计模式研究的开展和评价提供一个统一的参照系。

3.2 排除性描述的发掘

现有的设计模式描述中,大多仅讨论模式应该具有哪些成份、这些成份应该具有怎样的形态、成份之间应该具有怎样的关系。在这些正面约束中,往往隐含着一些排除性约束。例如要求类 A 使用类 B 有时就隐含着类 A 的使用者不能绕过类 A 使用类 B,同时也隐含着类 B 的相关方法不能使用类 A 的方法。在正向工程中,由于应用目的导向,模式的使用者通常不会违反这些隐含的排除性约束。但对于逆向工程而言,必须将排除性约束明确地表达出来和正面约束一起构成完整的识别条件。

设计模式是一个软件框架,它所提供的自由空间必须由正面约束和排除性约束来共同界定,如图 2 所示,因此,排除性约束是软件框架描述的重要手段。目前仅有个别研究孤立地给出了排除性约束,未与正面约束相联系,仅涉及语法约束,仅涉及模式内的成份。我们认为,排除性约束不应该孤立出现,该发掘它们和正面约束的内在关系,将它们整合在一个整体概念中;应该发掘语义层面的排除性约束。现有研究对那些处于模式之外,与模式有交互的成份没有任何约束,我们认为,虽然设计模式无法规定这些成份必须做什么,但是应该规定它们不能做什么(不能破坏模式结构和行为)。

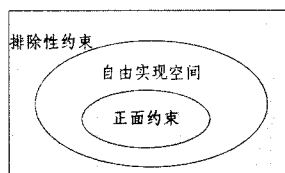


图 2 正面约束与排除性约束共同界定的设计模式自由实现空间

3.3 基于程序静态分析的设计模式语义特征识别

设计模式的很多核心概念——如“组合”、“适配”、“委

托”、“备忘”、“装饰”等含有大量抽象程度较高,内含丰富的语义。这些概念容易意会,但现有研究尚未提供有效方法以准确描述其中的语义。无论怎样抽象的语义最终都必须通过控制流和数据流来实现。我们发现,设计模式的语义特征大多可以通过控制流和数据流层面上的基本的静态语义关系得到精确描述,并通过传统的静态语义分析技术得到比较准确的识别。例如,设计模式中广泛存在多对一关联。我们发现,对象聚集与其拥有者是实现设计模式中多对一关系的基本形态,与之相关的角色行为也往往表现为聚集管理操作,如添加、删除、遍历等。不同模式中的聚集和聚集管理操作还具有不同的特征。现有研究仅能识别由数组和预定义的容器类实现的聚集,不能识别更为广泛的由链表实现的聚集。我们在研究中发现,通过别名分析、堆分析等传统静态语义分析技术可以有效区分具有同样数据结构的聚集关联和普通关联,表达和识别不同模式中的聚集关系和聚集管理操作,从而识别相应的模式成份。我们的研究表明,通过指针分析可以识别在设计模式中大量存在的使用指针实现的角色数据及相关行为。

每个设计模式都需要通过各成份的相互协作实现一定的语义,不同协作过程产生不同的数据流,这种语义特征可以通过数据依赖关系来表达。依赖性分析对于识别模式语义具有普遍意义。我们发掘了模式中各成份间存在的依赖关系、角色行为本身的依赖关系特征以及同一模式成份的不同风格的实现所具有的不同依赖性特征。在使用其他特征识别出若干模式成份的基础上,通过成份间依赖关系可以识别更多相关成份,使识别结果更为完整。特别地,通过依赖性分析可以识别现有研究中均未涉及的角色数据,使识别结果更为细致、完整,使模式的边界更为明确。对实现风格的识别提供了关于模式实现的更多信息,使识别结果更为具体、丰富。我们在研究发掘了观察者模式、备忘模式和适配模式不同成员之间的数据依赖关系,不仅较为全面、准确地识别了所有角色行为,还可以采用统一的机制识别因应用而异的角色数据,如目标数据和观察结果等,以表明该方法的有效性。

我们将指针分析和数据依赖性分析等传统分析技术引入设计模式语义特征的识别,有效地在底层语义分析技术和设计模式中的高层设计概念之间架起桥梁,成功地为这些高层概念提供了较为准确的语义特征描述和识别方法。设计模式中的语义层面很丰富,还需要将更多的传统语义分析技术引入模式语义描述和识别工作。传统语义分析技术对不同语言编写的不同程序具有普适性,以此为基础的识别技术也相应地具有较好的通用性。

模式的识别总体上可以分为两步进行:首先是根据模式的语法特征识别出候选者,然后在候选者中根据语义特征确定最终结果。此处的语义分析不是全局的,而是基于具体目的、步进式的。因此,有必要在传统的全局静态分析技术基础上,针对具体目的,研究新的步进式静态语义分析技术。传统的全局分析技术由于计算量太大,往往需要在精确性和效率之间进行折衷。我们相信,由于局部分析的计算量有限,精确的静态语义分析可以在设计模式识别中发挥有效作用。

3.4 可重构为设计模式的代码片断的识别

关于面向设计模式的手工重构已有一些实践总结^[27],但是还缺乏重构对象自动识别和重构流程方面的研究。可重构

为设计模式的代码在语法上基本没有约束,主要通过语义特征表达。识别这些代码必须首先对待识别程序进行语义分析,然后在分析结果中进行匹配。全局性语义分析工作量太大,考虑到设计模式的局部性,可以研究递增式的程序语义生成和匹配方法。

在此过程中,可以确定各数据和行为在设计模式中的对应角色,据此可以进一步研究面向设计模式的重构流程,并将其自动化。

3.5 设计模式的形式化描述技术

现有的形式化描述技术只能实现某一层面的描述,缺少综合描述的能力。我们认为可以采用当前发展较成熟的知识表示语言——描述逻辑作为设计模式形式化描述研究的基础。描述逻辑是一族知识表示语言,在框架和语义网的基础上增加形式语义,具有很好的可读性和易理解性;描述逻辑具有成熟的系统和优化的推理技术,这些都可以用于设计模式的识别和比较;描述逻辑以构造子为逻辑基础,具有很好的可扩展性,可根据不同的知识表示需要选用不同的具体描述逻辑。已有研究者提出了具有静态结构和动态行为的描述逻辑语言 ALC(T)^[2,32],它支持时间具体域 T 。还需要在该语言的基础上根据设计模式描述的需要增加新的构造子,并对描述时序行为的具体域 T 做进一步扩展。具体研究内容包括:

- 总结设计模式识别所涉及的基本语法和语义关系,将其作为描述逻辑中的基本概念,设计新的概念构造子表达高层设计概念并将其纳入描述逻辑知识库;利用描述逻辑知识库中的概念公理来构建设计模式的概念体系,使该描述体系具备可扩展性,可以表达今后发现的新模式。

- 对于动态行为中的时序关系,通过具体域连接子,将表示行为概念的时间属性投影到具体域中,并利用具体域上的时序谓词表示相应时间关系。

- 将设计模式中所涉及的不同层面的特征整合在描述逻辑概念体系中,特别是整合结构描述和行为描述,整合行为效果与行为的实施的描述。

在形式化描述研究中应注重发掘在非形式化描述中容易被忽视的内部一致性约束条件,研究其验证方法。

设计模式的形式化描述将为设计模式识别提供精确的识别目标,设计模式的识别效果也可以验证这种描述的有效性和正确性。在设计模式的形式化描述研究的基础上,可以进一步研究由描述驱动的自动识别。在形式化描述研究的基础上,可以进一步研究由描述驱动的自动识别。

结束语 设计模式自动识别研究有助于挖掘设计模式的内涵,有助于理解和分析遗产软件,有助于在再工程阶段发挥设计模式的作用。相关研究已开展十年有余,本文比较系统地总结和分析了现有研究中所存在的问题,即:(1)对识别目标缺乏统一、明确的界定;(2)缺乏严格定义和准确识别语义特征的能力;(3)框架表达能力不足;(4)对识别特征缺乏通用的形式化描述。这些问题最终导致识别不够准确、细致,识别方法缺乏通用性和灵活性。针对以上问题,本文提出了针对性的方案,即通过(1)层次化识别目标的提出、(2)排除性约束的发掘及与正面约束的整合、(3)基于静态分析的语义特征定义与识别、(4)基于描述逻辑的形式化描述技术等方面的研究来解决以上问题,给出了基本的研究思路。我们已经遵循这

个框架使用数据依赖性和堆分析技术成功地识别了部分模式的主要特征,由此,我们认为这个研究框架是基本合理和可行的。

参考文献

- [1] Erich G, et al. Design Pattern Elements of Reusable Object-Oriented Software[M]. New York: Addison-Wesley, 1995
- [2] Lutz C, Haarslev V, Moller R. A Concept Language with Role-forming Predicate Restrictions[R]. FBI-HH-M-276/97, University of Hamburg, Computer Science Department, Hamburg, 1997
- [3] Philippow I, Streitferdt D, Riebisch M, et al. An Approach for Reverse Engineering of Design Patterns [J]. Software and Systems Modeling, 2004, 4(1): 55-70
- [4] Huang Heyuan, Zhang Shensheng, Cao Jian, et al. A Practical Pattern Recovery Approach Based on Both Structural and Behavioral Analysis [J]. The Journal of Systems and Software, 2005, 75(1/2): 69-87
- [5] Jarvinen H, Kurki-Suonio R, Sakkinen M, et al. Object-oriented Specification of Reactive Systems [C]//Proceedings of the 12th International Conference on Software Engineering. Los Alamitos: IEEE Computer Society Press, 1990: 63-71
- [6] Allen J F. Towards a General Theory of Action and Time [J]. Artificial Intelligence, 1984, 23(2): 123-154
- [7] Nguyen-Cong V, Wang Yingxu. Specification of Design Patterns Using Real-time Process Algebra [C]//Proceedings of Canadian Conference on Electrical and Computer Engineering. Canada: IEEE CA, 2004, 3: 1545-1548
- [8] Wang Yingxu. The Real-Time Process Algebra (RTPA) [J]. Annals of Software Engineering, 2002, 14(1/4): 235-274
- [9] Hoare C A R. Communicating Sequential Processes [M]. New Jersey: Prentice Hall International, 1985
- [10] Soundarajan N, Hallstrom J O. Responsibilities and Rewards: Specifying Design Patterns [C]//Proceedings of the 26th International Conference on Software Engineering. Washington: IEEE Computer Society, 2004: 666-675
- [11] Costagliola G, De Lucia A, Deufemia V, et al. Design Pattern Recovery by Visual Language Parsing [C]//Proceedings of the Ninth European Conference on Software Maintenance and Re-engineering. Washington: IEEE Computer Society, 2005: 102-111
- [12] Wang Wei, Tzerpos V. Design Pattern Detection in Eiffel Systems [C] // Proceedings of the 12th Working Conference on Reverse Engineering. Washington: IEEE Computer Society, 2005: 165-174
- [13] Kramer C, Prechelt L. Design Recovery by Automated Search for Structural Design Patterns in Object-oriented Software [C] //Proceedings of Working Conference on Reverse Engineering. Washington: IEEE Computer Society, 1996: 208-215
- [14] Brown K. Design Reverse - engineering and Automated Design Pattern Detection in Smalltalk [D]. USA: North Carolina State University, 1996
- [15] Seemann J, von Gudenberg J W. Pattern-based Design Recovery of Java Software [C] //Proceedings of the 6th ACM SIGSOFT International Symposium on Foundations of Software Engineering. New York: ACM Press, 1998: 10-16

- [16] Heuzeroth D, Holl T, Hogstrom G, et al. Automatic Design Pattern Detection [C] // Proceedings of the 11th International Workshop on Program Comprehension. Washington; IEEE Computer Society Press, 2003; 94-103
- [17] Högström G. Detecting Communication Design Patterns by Both Static and Dynamic Analyses [D]. 03016. ISSN 1650-2647. School of Mathematics and Systems Engineering, Växjö University, April 2003
- [18] Wendehals L. Improving Design Pattern Instance Recognition by Dynamic Analysis [C] // Proceedings of the International Conference on Software Engineering. Workshop on Dynamic Analysis. Washington; IEEE Computer Society, 2003; 29-32
- [19] Smith J M C, Stotts D. SPQR: Flexible Automated Design Pattern Extraction from Source Code [C] // Proceedings of the 18th IEEE International Conference on Automated Software Engineering. Washington; IEEE Computer Society, 2003; 215-224
- [20] Smith J M C, Stotts D. Case Studies in Automated Design Pattern Detection in C++ Code using SPQR[R]. TR05-013. Department of Computer Science, Univ. of North Carolina at Chapel Hill. June 2005
- [21] Nickel U, Niere J, Zündorf A. The FUJABA Environment [C] // Proceedings of the 22nd International Conference on Software Engineering. New York; ACM, 2000; 742-745
- [22] Arcelli F, Masiero S, Raibulet C, et al. A Comparison of Reverse Engineering Tools Based on Design Pattern Decomposition // Proceedings of the 2005 Australian Software Engineering Conference. Washington; IEEE Computer Society, 2005; 262-269
- [23] Antoniol G, Casazza G, Penta M D, et al. Object-oriented Design Pattern Recovery[J]. The Journal of System & Software. 2001, 59(2); 181-196
- [24] Niere J, Schafer W, Wadsack J P, et al. Towards Pattern-Based Design Recovery [C] // Proceedings of International Conference on Software Engineering. New York; ACM, 2002; 338-348
- [25] Eide E, Reid A, Regehr J, et al. Static and Dynamic Structure in Design Patterns [C] // Proceedings of the International Conference on Software Engineering. New York; ACM, 2002; 208-218
- [26] Derrick J, Boiten E. Refinement in Z and Object-Z; Foundations and Advanced Applications [M]. UK; Springer-Verlag London Limited, 2001
- [27] Kerievsky J. Refactoring to Patterns [M]. US; Pearson Education Inc. , 2005
- [28] Dietrich J, Elgar C. A Formal Description of Design Patterns Using OWL [C] // Proceedings of the 2005 Australian Software Engineering Conference. Washington; IEEE Computer Society, 2005; 243-250
- [29] France R B, Kim Dae-Kyoo, Ghosh S, et al. A UML-Based Pattern Specification Technique [J]. IEEE Transactions on Software Engineering, 2004, 30(3); 193-206
- [30] Kim Soon - Kyeong, Carrington D A. Using Integrated Meta-modeling to Define OO Design Patterns with Object-Z and UML [C] // Proceedings of the 11th Asia-Pacific Software Engineering Conference. Washington; IEEE Computer Society, 2004; 257-264
- [31] Mitchell R, Mckim J. Design by Contract [M]. US; Addison-Wesley, 2002
- [32] Gabbay D M, Hodkinson I M, Reynolds M A. Temporal Logic: Mathematical Foundations and Computational Aspects [M]. Volume 1. USA; Oxford University Press, 1994
- [33] Pettersson N. Measuring Precision for Static and Dynamic Design Pattern Recognition as a Function of Coverage [C] // Proceedings of the Third International Workshop on Dynamic Analysis. New York; ACM, 2005; 1-7
- [34] Heuzeroth D, Mandel S, Lowe W. Generating Design Pattern Detectors from Pattern Specifications [C] // Proceedings of the 18th IEEE International Conference on Automated Software Engineering. Washington; IEEE Computer Society, 2003; 245-248
- [35] Baader F, Calvanese D, McGuinness D L, et al. The Description Logic Handbook; Theory, Implementation, and Applications [M]. UK; Cambridge University Press, 2003
- [36] Collins A M, Quillian M R. Retrieval Time from Semantic Memory [J]. Journal of Verbal Learning and Verbal Behavior. 1969, 8(2); 240-247. Reprinted in Computation and Intelligence. Edited by G. F. Luger. Menlo Park, CA; AAI Press; 191-201
- [37] 贾育, 顾毓清. 基于领域特征空间的构件语义表示方法[J]. 软件学报, 2002, 13(2); 311-316
- [38] Jezequel J M, Train M, Mingins C. Design patterns and contracts [M]. USA; Addison-Wesley, 1999
- [39] Heuzeroth D, Holl T, Löwe W. Combining Static and Dynamic Analyses to Detect Interaction Patterns // IDPT'02. Pasadena, CA

(上接第 84 页)

结束语 本文提出了一种新的基于最小 ID 的改进算法, 算法兼顾最小 ID 算法成簇简单的优势, 引入异常度的概念, 在网络拓扑变化时根据异常度做簇结构调整, 通过改进簇结构变化时调整机制来达到提升分簇结构稳定性, 减少簇结构变化远距离簇间路由发现时所需时间及开销。算法性能分析和实验表明了改进算法簇头平均更换次数少, 平均担任时间长, 簇结构稳定性得到提升。

参 考 文 献

- [1] 王金龙, 等. Ad Hoc 移动无线网络[M]. 北京: 国防工业出版社, 2002
- [2] Chiang C-C, Wu H-K, Liu W, et al. Routing in Clustered Multi-hop, Mobile Wireless Networks with Fading Channel [A] // Proceedings of SICON. 1997
- [3] Lin C R, Gerla M. Adaptive Clustering for Mobile Wireless Networks [J]. IEEE Journal on Selected Areas in Communications, 1997, 15(7)
- [4] Yu J, Chong P. A survey of clustering schemes for mobile Ad hoc networks[J]. IEEE Communications Surveys, 2005, 7(1); 32-48
- [5] Basagni S. Distributed Clustering for Ad Hoc Networks [A] // Proc. of Int'l Symp. on Parallel Architectures, Algorithms and Networks[C]. 1999; 310-315
- [6] Basu P, Khan N, Little T D. A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks [A] // Proc. of IEEE ICDCS 2001 Workshop on Wireless Networks and Mobile Computing [C]. 2001; 413-418
- [7] 程伟明, 周新运. 一个用于 Ad Hoc 网络的分簇方法[J]. 计算机学报, 2005, 5(28); 864-868
- [8] 王海涛, 郑少仁, 刘晓明. 移动 Ad hoc 网络中的分簇算法[J]. 解放军理工大学学报, 2004, 5(3); 28-32
- [9] 徐浩, 慕德俊, 李立欣. 一种 Ad hoc 网络按需式分簇路由算法[J]. 计算机工程与应用, 2007, 43(14); 3-6