

基于 MDA 的需求捕获工具的设计与实现

曾 一^{1,2} 黄兴砚¹ 李函逾¹ 王翠钦¹

(重庆大学计算机学院 重庆 400044)¹ (软件理论与技术重庆市重点实验室 重庆 400044)²

摘 要 目前 MDA 开发过程中仍采用人工捕获、文字描述的传统需求获取形式,这影响了需求模型与 PIM 模型的精度和一致性,降低了 MDA 开发的自动化程度。重点研究开发了一个可视化需求捕获工具。该工具采用 MDA 框架以及 GEF 等技术进行开发,基于目标-情景的捕获方式进行需求获取,同时支持以需求文档与 XML 的形式将需求模型导出,为需求模型到 PIM 模型的转换提供了足够信息。最后通过实验验证了该捕获工具的有效性。该工具弥补了 MDA 中没有独立需求捕获阶段的不足,在一定程度上完善了 MDA 的开发流程,提高了 MDA 软件开发的自动化程度。

关键词 模型驱动架构,需求捕获,需求元模型,EMF,GEF

中图法分类号 TP311.5 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.10.044

Design and Implementation of Requirements Capture Tools Based on MDA

ZENG Yi^{1,2} HUANG Xing-yan¹ LI Han-yu¹ WANG Cui-qin¹

(College of Computer Science, Chongqing University, Chongqing 400044, China)¹

(Key Laboratory of Software Theory and Technology in Chongqing, Chongqing 400044, China)²

Abstract By now, the traditional ways of MDA developing process have always been the manual capture and text descriptions, which has affected the accuracy and consistency of both demand model and PIM model, and reduced the automation degree of MDA development. This article offered the way of how to develop a visual demand capture tool. With the technology of MDA framework and the GEF adopted in whose development, this tool runs the demand capture in the goal-situation capturing way. Whereas it offers the feasibility of exporting the demand model in the form of both requirements document and XML, and provides enough information for the conversion from demand model to PMI model. Finally, the examples offered in this article demonstrate the effectiveness of the capture tool. By this tool, the lack of independent requirements capture process can be made up, and to some extent, the development of MDA process and the automation degree of MDA software can be improved.

Keywords MDA, Requirement capture, Requirements metamodel, EMF, GEF

OMG(Object Management Group)提出基于 UML、XMI、MOF、CWM 等工业标准的模型驱动架构(Model Driven Architecture, MDA)^[1],其核心思想是使具体实现与平台和技术无关,从而提高软件开发的可重用性以及开发效率,通过模型的定义、模型到模型的转换以及模型到代码的转换,使整个软件开发更为自动化。

如图 1 所示,目前 MDA 并没有自己独立的需求阶段,仍采用人工捕获和文字描述的传统方式,由于人的主观性因素常常导致同一需求产生不同的需求文档,影响了需求模型与 PIM 模型的精度和一致性。而需求的人工捕获与需求模型、PIM 模型的人工建立,导致需求捕获与 MDA 自动转换的开发过程脱节。已有支持 MDA 需求捕获的工具^[2,3]采用传统软件开发模式完成,以 CRC 卡填写需求表格形式完成需求捕获,其捕获工具自动化程度不高,不能很好地满足 MDA 开发

的需求,填表的形式使专业人员与客户在需求捕获过程中沟通困难,不利于需求的捕获,捕获工具的界面也不够友好。鉴于以上问题,本文将需求过程引入 MDA 开发流程中,建立一个基于 MDA 的可视化图形界面的需求捕获工具。该工具可以使用户在基于目标-情景方式的规则下获取需求^[4],并将获取的需求以文档与 XML 的形式导出,使需求模型有更好的精度和一致性,降低了人为因素的影响,为将来需求模型到 PIM 模型的转换提供了基础和足够的信息,使捕获方式更加友好,软件开发的自动化程度进一步提高,在一定程度上完善了 MDA 的开发流程。

图 2 给出了结合 MDA 的需求捕获过程图,本文将要实现的需求捕获工具的主要功能如图中虚线框中所示,它总共包含了以下几部分:1)研究现有的需求捕获方式和过程,采用基于目标-情景的需求捕获方式;2)通过 MDA 元建模技术采

到稿日期:2013-11-05 返修日期:2014-01-19 本文受国家自然科学基金(61272194)资助。

曾 一(1961—),男,教授,CCF 会员,主要研究领域为软件工程理论及应用;黄兴砚(1987—),男,硕士生,主要研究领域为软件工程理论,E-mail:huangxingyan1987@sina.com;李函逾(1991—),男,硕士生,主要研究领域为软件工程理论及应用;王翠钦(1988—),女,硕士生,主要研究领域为软件工程理论及应用。

用 EMF (Eclipse Modeling Framework) 框架^[5] 建立符合 MDA 标准的需求元模型; 3) 通过需求元模型建立相应的需求捕获工具, 捕获需求信息生成相应需求模型, 实现需求捕获功能; 4) 研究 M2T (Model To Text) 技术^[6], 通过 Acceleo 模板技术设计需求模型到需求文档以及 XML 转换的模板文件, 支持需求文档以及 XML 文件的导出。

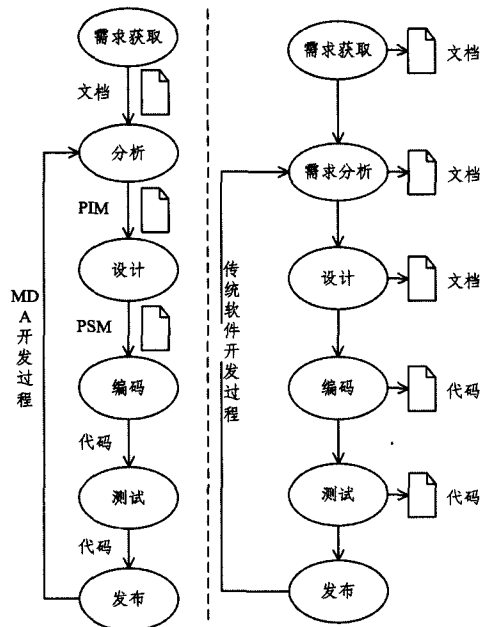


图1 MDA与传统开发过程

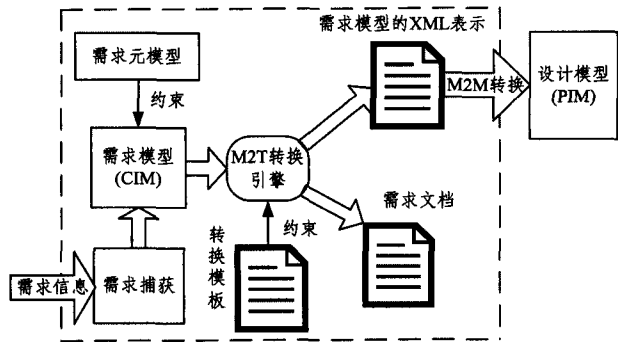


图2 结合MDA的需求捕获过程

1 需求捕获

1.1 需求捕获及其相关方法

需求捕获是需求工程的一部分, 它是一个从人、文档或者环境中收集与软件系统相关的、来自不同来源信息的过程, 其目的是从项目的规划开始建立最初的原始需求。

这个活动看似是简单的信息传递、转换并记录的过程: 需求人员从用户获取信息并以特定形式记录下来。实际上, 它存在着需求随着环境、时间改变而变化; 需求的获取也由于需求人员和用户的主观性容易产生偏差; 多个用户的需求还可能蕴含矛盾等问题。目前, 为了解决需求人员捕获用户需求时所遇到的问题, 已存在多种需求捕获方式, 其中主要有:

面向目标的方法^[7,8]; 其是上世纪末在需求工程领域中逐渐兴起的, 它通过一个元模型来定义需求捕获中所需收集的信息内容。其优点是保证了需求的完整性, 避免了无关需求, 使目标层次更加结构化; 缺点是目标的发现困难。

面向方面的方法; 其主要思想是通过引入方面的概念, 分离关注点, 减少代码间的耦合, 提高软件的适应性、可扩展性和维护性。但是识别其横切仍然是一个难题。

基于情景的方法^[9]; 其在需求工程领域得到了广泛的应用, 它着眼于待开发系统的具体实现细节, 从具体情景实例出发, 便于需求人员和用户进行交流。其本质上是局部的, 很难在一系列的情境中保持一致性和完整性, 而保持合适的粒度也显得十分困难。

1.2 基于目标-情景的需求捕获方法

本文在研究多种需求捕获方式的优缺点后, 采用了基于目标-情景的需求捕获方法^[4]。它有效地结合了基于目标和基于场景的需求捕获方法, 解决了基于目标方法中目标发现困难的问题以及基于情景方法中保持一致性、完整性的问题, 克服了两者的不足和局限, 并提供了一套完善的目标发现策略。在需求捕获过程中, 基于目标-情景的需求捕获将目标看作是情景的上下文结构, 通过目标建立情景结构, 在建立情景结构的同时通过分析来确认已实现的预期目标以及发现新目标。其流程如图3所示。

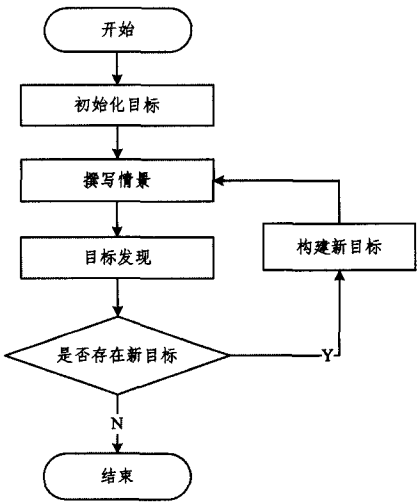


图3 需求捕获流程

基于目标-情景的捕获方法是一个目标发现→情景撰写→目标发现的循环需求捕获过程, 其通过3个步骤循环来完成需求的捕获:

- (1) 需求捕获开始时, 先需要抽取初始化目标, 以确定待实现系统的最高层次目标, 并为其撰写相应的情景;
- (2) 通过(1)将得到一个完整的需求, 并通过分析情景事件流, 发现并确定新的目标;
- (3) 重复(1)、(2), 通过目标发现和情景撰写过程不断挖掘新的目标直到系统的需求获取完成。

2 需求元模型的建立

分析基于目标-情景需求捕获方式的需要, 在目标-情景需求捕获原有基础上, 通过MDA元建模技术采用EMF框架建立符合MDA标准的需求元模型, 以支持MDA需求捕获工具的建立, 为需求捕获提供保证。

2.1 需求元模型

目前许多文献^[10,11]都提出了基于不同需求捕获方式的需求元模型, 它们之间因捕获方式以及侧重点的不同, 存在着

中。事件流分为 4 种类型:顺序、迭代、条件、并发,其中迭代和条件都需要进行条件语句,这些条件语句也包含了参数 SPParameter。

3.1 Ecore 模型

```

classDiagram
    class Requirement
    class Scenario
    class Verb
    class Goal
    class Parameter
    class State
    class Action
    class AtomicAction
    class FlowAction
    class Agent
    class Resource
    class FlowCondition
    class SParameter
    class Constraint
    class Sequence
    class Iteration
    class Concurrency
    class Entity
    class Direction
    class Way
    class Stakeholder
    class Destination
    class Source
    class Means
    class Manner

    Requirement "1" -- "1" Scenario
    Requirement "1" -- "1" Verb
    Requirement "1" -- "1" Goal
    Requirement "1" -- "1" Parameter
    Scenario "1" -- "1" State
    Scenario "1" -- "0..1" Action
    Action <|-- AtomicAction
    Action <|-- FlowAction
    AtomicAction "1" -- "1" Agent
    AtomicAction "1" -- "1" Resource
    FlowAction "1..*" -- "1" FlowCondition
    FlowAction "1" -- "1" Iteration
    FlowAction "1" -- "1" Concurrency
    FlowCondition "1" -- "0..1" SParameter
    FlowCondition "1" -- "1" Constraint
    FlowCondition "1" -- "1" Sequence
    Parameter <|-- Entity
    Parameter <|-- Direction
    Parameter <|-- Way
    Parameter <|-- Stakeholder
    Direction <|-- Destination
    Direction <|-- Source
    Way <|-- Means
    Way <|-- Manner
  
```

2.1.1 目标元模型结构

Entity:指定目标中所包含的实体,具体有两种情况:一种情况表示在目标被实现之前假定存在的实体;另一种情况表示目标完成之前不存在的实体或者抽象的实体。

Direction:表示通信对象的起始位置和目标位置,Direction 有两种类型:代表起始位置的 Source 和代表目标位置的 Destination。

Way:表示完成动作所使用的方式或手段,具有 Means 和 Manner 两种类型。其中 Means 是指一个用于帮助完成目标的实体,Manner 则表示一个复杂的方式或手段,其中可能包含另一个目标。

Stackholder: 利益关系人, 是指目标完成时受益方是谁。

2.1.2 情景元模型结构

情景是指为完成一个目标,几个代理之间进行的有目的、有意义的交互。一个情景包含前置状态 `preState` 和后置状态 `postState`。前置状态定义了情景触发的前置条件。后置状态定义了情景完成时情景所达到的状态。多个情景组合起来就可以构成一个复杂系统。

情景是由一系列事件 Action 构成的,事件描述了从初始状态到最终状态的事件路径。事件 Action 是由原子事件 AtomicAction 或者事件流 ActionFlow 组成的。原子事件表示从一个代理到另一个代理的信息通信,其信息中包含了一些参数 SParameter,这些参数可能参与到其他的原子事件当

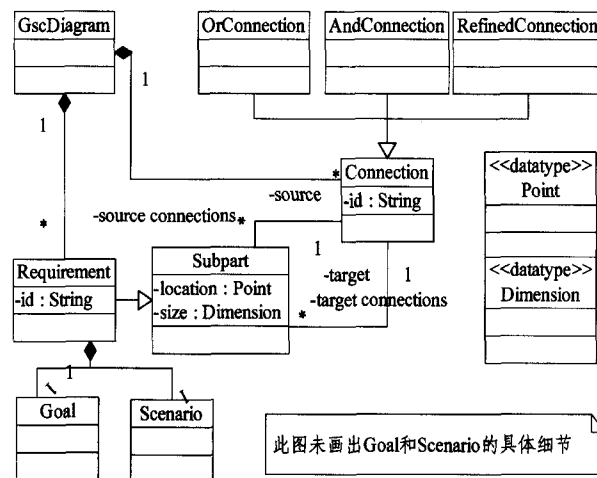


图 5 Ecore 模型

与图 2 相比,改进后的元模型新增了部分元素,包括 GscDiagram、Subpart、Connection 及其子类以及自定义的 Point 数据类型和 Dimension 数据类型。

GscDiagram;它表示需求捕获时的根元素,在通过工具进行可视化图形捕获需求时,必须先建立一个 GscDiagram 画布,才能在这块画布上添加图形元素,如建立需求 Requirement、引用关联等。

Subpart:它是一个抽象类,是需求 Requirement 的父类,属性 location 和 size 分别表示元素在 GscDiagram 中的位置和大小,其中 location 和 size 都存在于自己的数据格式:Point 和 Dimension。

Connection:它是一个用以表示关联的抽象类,能够对关联进行动态操作和修改,并实时保存联系两端的 Subpart 子类。它有 3 个子类: AndConnection、OrConnection、Refined-Connection,分别用来表示需求 Requirement 之间的 AND 关系、OR 关系以及 Refined 关系。

Requirement:它与图 3 一致,与 Goal 和 Scenario 具有严格的控制关系,即一个 Requirement 中只能拥有一个 Goal、Scenario 对。

经过扩展后,在 EMF 下利用 ECore 建立符合 MOF 标准的需求元模型如图 6 所示。

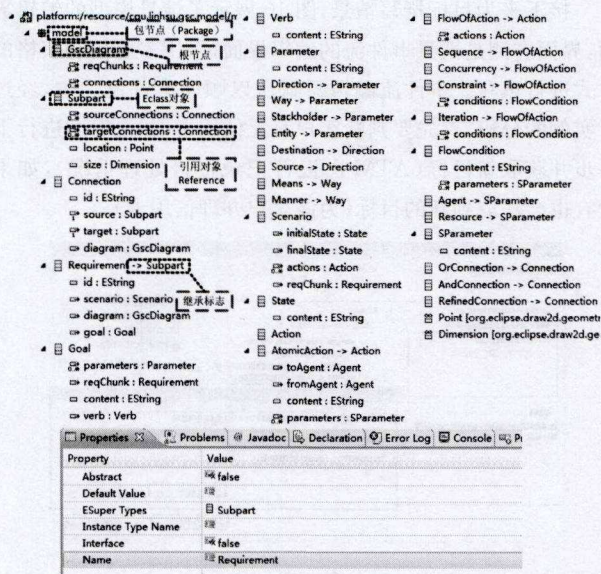


图6 需求元模型 Ecore 建模

扩展后建立的 Ecore 模型是完全根据基于目标-情景捕获方式的捕获功能进行建立的,包含了其捕获方法所有功能的元素,能很好满足基于目标-情景捕获的需要。

3.2 模型到代码的转换

3.2.1 模型到代码的转换

通过本文设计的工具捕获需求后,将需求模型转换成需求文档以及 XML 文件并导出,通过 XML 的导出为将来需求模型到设计模型的转换提供了保证。在该转换过程中,将采用 Eclipse 平台下 M2T 项目中的 Acceleo 开发框架^[12]。图 7 给出了基于 Acceleo 模板技术的模型到代码的转换过程,主要包含以下 3 步:1)分析需求元模型结构,根据将要导出的文档要求以及需求元模型结构,编写合适的 Acceleo 模板文件;2)根据需求元模型展开需求捕获活动,得到需求模型,并生成相应的 XML 模型文件;3)运行 Acceleo 代码生成器,读入 XML 源文件以及转换模板文件,最终得到目标文件。

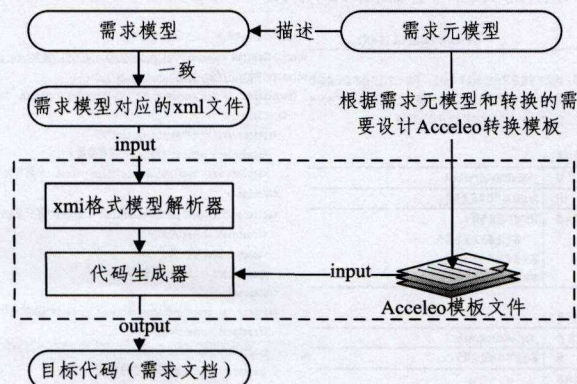


图7 模型到代码转换过程

3.2.2 Ecore 元模型 XML 文件的分析

EMF 在持久化过程中通过 XMLSchema 格式来存储 Ecore 元模型。其主要包含 XML 版本、编码信息、定义 XML 格式的头文件引用和 ecore 模型的元素节点信息。通过对 Schema 的分析,在模型到代码的转换过程中提取 EClass、EAttribute、EReference 和 EDataType 元素信息,将其注入到模板文件中,能自动完成从需求模型到需求文档的转换。

3.2.3 Acceleo 模板文件设计

本文采用符合 MOF 规范的 Acceleo 工具制作出对应的模板文件实现模型到代码的转换(需求模型 XML 到需求文档)。其中 Acceleo 的语法比较简单,主要包含常用语法与 Acceleo 模板提供的基本服务两个部分。常用语法包括 6 种: comment 注释、metamodel、import、script、for 以及 if 语句。模板服务包括 9 种:上下文服务、类型服务、对象服务、属性服务、需求服务、资源服务、专用服务、字符串服务和 xpath 服务。在模型到代码的转换中首先导入需求元模型,接着设置根元素,从根元素为起始点开始循环查找子元素,并输出需求中的目标内容和情景内容(包括事件流)。图 8 给出了本文需求元模型结构的部分模板文件结构。

```

[comment encoding = UTF-8 /]
[**导入依赖的需求元模型/]
[module generate('http://cqu.linhxu.gsc.model/1.0')]
[**设置根元素/]
[template public generateElement(aGscDiagram : GscDiagram)]
[comment @main/]
[file ('requirement.txt', false, 'UTF-8')] [**设置导出文件名/]
[**按模板格式,输出需求中的内容/]
[for (it : Requirement | aGscDiagram.requirements)]
    需求 ID [it.id.toString() /] : [**需求Id/]
    目 标
    [it.goal.content.toString() /] : [**目标内容/]
    情景内容
    [for (action : Action | it.scenario.actions)]
        [**输出原子事件内容/]
        [if (action.eClass().name.equalsIgnoreCase('AtomicAction'))]
            [action.current(AtomicAction).content/]
        [/if]
        [**输出条件事件流内容/]
        [if (action.eClass().name.equalsIgnoreCase('Constraint'))]
            [action.current(Constraint).conditions.content/]
            [for (action : Action | action.current(Constraint).actions)]
                [action.current(AtomicAction).content/]
            [/for]
        [/if]
    [/for]
[/for]
[/file]
[/template]
  
```

图8 模板文件片段

4 需求捕获工具的实现

4.1 需求捕获工具的实现

结合 Eclipse 下的 EMF、GEF 以及 RCP 框架,实现一个基于目标-情景的可视化需求捕获工具界面布局以及工具组件。

4.1.1 实现需求捕获工具技术以及模型的图形化

首先,EMF 提供了一个模型编辑器,它采用了一种基于树状结构的模型编辑方式,在复杂的模型结构下,各结点之间关系不明显,容易导致错误。本文将 GEF 框架与 RCP 框架结合 EMF 建立一个具有图形化的模型编辑器,并构建、部署富客户端程序的平台。

回顾 3.1 节中扩展后的需求元模型,在实现需求捕获工具时需要将这些元模型元素变为相应的图形元素,需求人员就可以利用这些图形进行建模。这里需要将 Ecore 中的 Requirement、GscDiagram 以及需求之间的 3 种联系 RefinedConnection、AndConnection、OrConnection 以图形元素的形式提供给需求人员,而 Goal 和 Scenario 及其子类内嵌在 Requirement 模块中,以表格形式支持目标与情景模块的编辑。

4.1.2 GEF 模型编辑器的实现

GEF 模型编辑器的实现可为用户提供图形化的工作平

台,其中包括面板区域和编辑区域。用户可以拖动面板区域的图形元素至编辑区域,生成相应的图形,并根据实际需要对方图形元素进行编辑操作。GEF 是将模型(Model)与图形元素(View)通过 EditPart 类(Controller)进行关联,EditPart 可以存在子 Editpart,它们之间的继承关系与模型间的关系一致。GEF 提供两种 EditPart 的实现:GraphicalEditPart 和 Tree-EditPart。GraphicalEditPart 使用 Figure 作为视图,支持连接;TreeEditPart 则使用 SWT 中的 TreeItem 作为视图,用于显示树形结构。本文采用前者实现图形的创建、连接以及编辑操作。

当用户进行交互操作,模型发生变化时,采用 EMF 提供的更新机制来通知视图进行更新。图形的更新在 EditPart 的 refreshVisual 函数中完成。

另外,在初始化操作中,需要读取外部文件的位置,如果文件中包含有内容,则需要读取其中内容,并建立相应的模型;如果没有,则建立一个根模型 GscDiagram,最后编辑器设置所依赖的根元素为 GscDiagram。

在保存当前建模内容时,需要调用 doSave 方法与 EMF 提供的相关函数进行操作,实现需求模型对应的 XML 文档的导出。

4.1.3 RCP

上一节中,模型编辑器不能离开父窗体独立执行,许多窗体工件如菜单栏、工具栏等还没有实现,这里将使用 RCP 框架。RCP 框架提供了许多功能,其中包括窗口、透视图、视图、编辑器、动作集以及独立运行等。

通过 Eclipse 平台下 PDE 向导提供的 RCP 模板,可以生成一个应用程序 Application。原始的 Application 只提供了一个视图窗口,不包含其他任何工具组件。这里使用 ApplicationActionBarAdvisor 类实现透视图以外的菜单栏、工具栏、状态栏等组件;Perspective 类完成透视图窗口的整体布局。

5 基于 MDA 的需求捕获工具的应用实例

下面将以自动取款机 ATM 提供取款服务为例,介绍该工具的使用过程。首先以基于目标-情景需求获取方法的流程,建立一个新的需求块,目标内容为“使用 ATM 为客户提供取款服务”。分析目标内容发现,ATM(源 source)为客户(目标 destination)提供(动作 verb)取款服务,提供的现金成为实体 entity,采用 ATM 提款而非柜台取款表现了实现的手段(means),如图 9 所示。

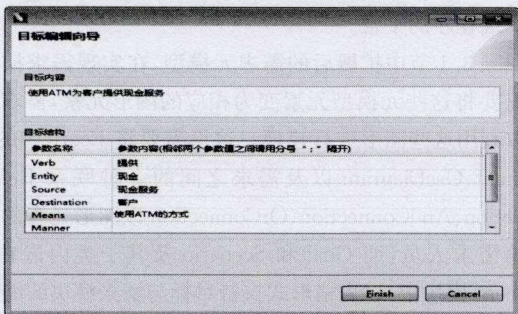


图 9 目标编辑向导

接下来为目标撰写情景,图 10 展示了情景模型的编辑界面,界面右侧是情景事件流的编辑界面,采用多路树和表格的方式进行填写。事件流编辑器默认以顺序事件流为主,经过对实例的分析,首先要判断客户有无信用卡,如果有则进行下一步并获取新目标(ATM 向银行记录报告处理情况),如果没有也会获取到新的目标(为该客户办理信用卡)。

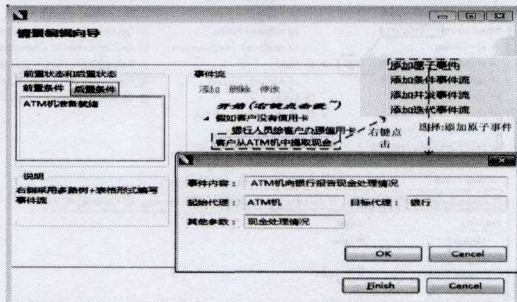


图 10 情景编辑向导

通过分析情景事件流并采用基于目标-情景需求捕获方法的需求捕获规则,最终可以得到 3 个新目标,生成 3 个子 Requirement,并用 AND 关系,将其关联起来,如图 11 所示。

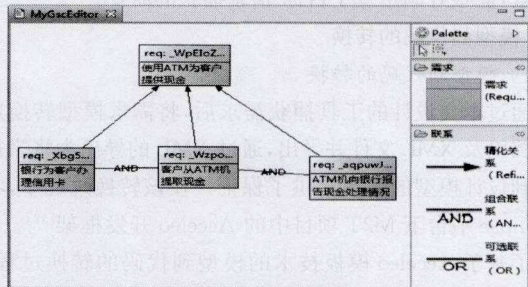


图 11 需求之间的联系

通过基于目标-情景的捕获方法,建立完整的需求模型。需求模型可以通过工具的菜单导出功能导出 XML 和需求文档,导出的文档内容如图 12 所示。

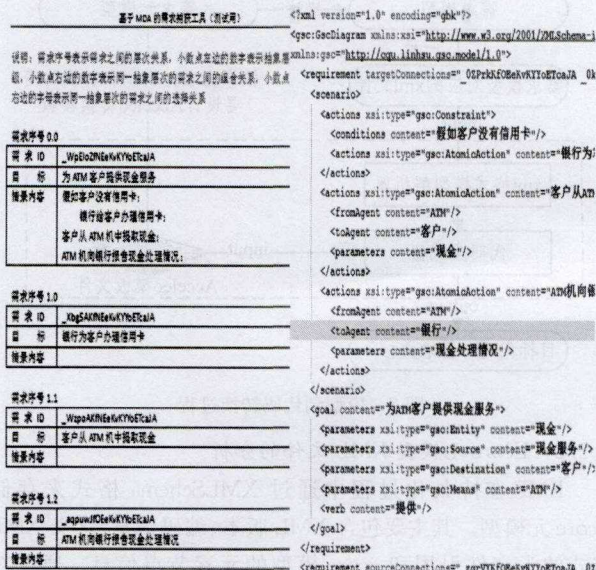


图 12 需求之间的联系

结束语 本文通过对需求捕获方法的研究,分析现有的

需求工具的不足,提出了一种图形化的需求捕获工具。用户依照基于目标-情景的捕获方式规则能有效地捕获需求,使捕获的需求较为一致、精确,降低了人工捕获描述过程中主观因素造成的影响,并支持以 XML 和需求文档方式导出需求模型,为需求模型到 PIM 模型的转换提供了基础和足够的信息,较好地保证了模型的精度以及 MDA 自动化转换的顺利进行。由于没有进行较大规模的验证,需要通过实验进一步完善需求捕获元模型和 Ecore 模型,以满足更复杂的需要;进一步提高需求捕获的精度。

使用该工具捕获的需求模型,可通过形式化语言进行定义并可转换为 PIM 模型^[13]。该工具弥补了 MDA 中没有独立需求捕获阶段的不足,在一定程度上完善了 MDA 的开发流程,提高了 MDA 软件开发的自动化程度。

参 考 文 献

- [1] 陈平,王柏. MDA—新一代软件开发方法学的挑战与发展研究[J]. 计算机科学,2005(3):127-131
- [2] 宋晓玉. 基于 MDA 的需求工具的研究和设计[D]. 西安:西安电子科技大学,2008
- [3] 魏晖瑜. 基于 MDA 的需求自动建模工具的设计和实现[D]. 西安:西安电子科技大学,2010
- [4] 李勇华. 目标和场景相结合的需求工程[D]. 武汉:武汉大学,

(上接第 172 页)

同时实验也表明,在不同网络初始总能量环境下,DUBP 算法仍然具有更好的稳定性、能效性和适用性,能在一定程度上节省整个网络的能量消耗、优化节点负载均衡性,网络寿命得到明显提高。

参 考 文 献

- [1] 刘安丰,任炬,徐娟,等. 异构传感器网络能量空洞分析与避免研究[J]. 软件学报,2012,23(9):2438-2448
- [2] 刘唐,汪小芬,杨进. 基于相对距离的多级能量异构传感器网络成簇算法[J]. 计算机科学,2012,39(9):119-121
- [3] Petteri N. Reinforcement learning for routing in Ad Hoc networks[C]//Proceedings of the 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks. Limassol, Cyprus: IEEE,2007:1-8
- [4] Heinzelman W, Chandrakasan A, Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks [J]. IEEE Trans on Wireless Communications,2002,1(4):660-670

2006

- [5] Paternostro M. EMF; Eclipse Modeling Framework [M]. Addison-Wesley Professional
- [6] Eclipse Foundation. Model To Text(M2T)[OL]. <http://www.eclipse.org/modeling/m2t>
- [7] Dardenne A, van Lamsweerde A, Fickas S. Goal-directed requirements acquisition[J]. Science of Computer Programming,1993,20(1/2):3-50
- [8] Lapouchnian A. Goal-oriented requirements engineering: An overview of the current research[R]. University of Toronto,2005
- [9] 郑宇恒. 基于场景的软件需求建模研究[D]. 杭州:浙江师范大学,2009
- [10] Molina F, Pardillo J, Cachero C, et al. An MDE modelling framework for measurable goal-oriented requirements[J]. International Journal of Intelligent Systems,2010,25(8):757-783
- [11] Berre A J. Comet (component and model based development methodology)[OL]. <http://modelbased.net/comet/>
- [12] Obeo. Acceleo User Guide[OL]. <http://www.acceleo.org/doc/obeo/en/acceleo-2.6-user-guide.pdf>
- [13] 洪媛. 基于元模型的 MDA 软件建模技术研究[D]. 重庆:重庆大学,2009

- [5] Younis O, Fahmy S. HEED: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks[J]. IEEE Trans on Mobile Computing,2004,3(4):660-669
- [6] 陈庆章,赵小敏,陈晓莹. 提高无线传感器网络能效的双轮成簇协议设计[J]. 软件学报,2010,21(11):2933-2943
- [7] 刘明,曹建农,陈贵海,等. EADEEG: 能量感知的无线传感器网络数据收集协议[J]. 软件学报,2007,18(5):1092-1109
- [8] 周新莲,吴敏,徐建波. BPEC: 无线传感器网络中一种能量感知的分布式分簇算法[J]. 计算机研究与发展,2009,46(5):723-730
- [9] 廖鹰,齐欢,王晓红,等. 基于距离和分布的无线传感器网络分簇算法[J]. 华中科技大学学报,2012,40(6):34-38
- [10] Zhou H B, Y M, Hu Y Q, et al. A novel stable selection and reliable transmission protocol for clustered heterogeneous wireless sensor networks[J]. Computer Communications,2010,33(15):1843-1849
- [11] 苏光奎,李春葆. 数据结构导学[M]. 北京:清华大学出版社,2002