

基于问题框架的计算机辅助需求工程工具的研发

刘国源 万光海 庞柳 李智

(广西师范大学计算机科学与信息工程学院 桂林 541004)

摘要 问题框架(Problem Frames, PF)已经得到需求工程界的广泛重视和研究。目前,问题框架在理论基础和开发方法等方面已经有很多研究成果,但如何把它们应用到实际的软件开发项目中仍是一个有待解决的问题。本工具可帮助系统分析员实现从用户需求到软件规约的平滑和可推理的变换,从而为软件开发项目中需求的沟通、建模及分析等提供技术支持。此外,该工具产生的结果也为下一步的软件设计提供帮助。它对促进问题框架方法的进一步发展和成熟,如从研究走向实用,具有重要的推动作用。

关键词 问题框架,问题图,计算机辅助需求工程(CASE)

中图分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.11.027

Research and Development of Computer-aided Requirements Engineering Tool Based on Problem Frames

LIU Guo-yuan WAN Guang-hai PANG Liu LI Zhi

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

Abstract Problem Frames (PF) are widely regarded as an important research subject by the requirements engineering community. Currently, a lot of research work is done in the theoretical foundations and development methodologies of PF. However, how to apply them in the practice of a software development project remains a problem to be solved. A software prototype was developed to assist system analysts to derive software specifications from user requirements in a smooth and logical way, thus providing technical support for requirements communication, modeling and analysis in a software development project. Furthermore, the end products of the tool application also provide help for the software design in the development phase downstream. This software prototype will promote the further development of PF and drive it to maturity, i. e., from theoretical research to practical applications.

Keywords Problem Frames (PF), Problem diagram, Computer-aided Requirements Engineering (CARE)

1 引言

问题框架(Problem Frames, PF)是由软件工程著名学者 Michael A. Jackson 提出的用来帮助系统分析员分析和结构化软件开发问题的一种方法^[1,2]。该方法已经得到软件工程界的广泛重视和研究,例如从 2004 年至 2010 年,国际软件工程大会(ICSE)每隔一届都会举办一次问题框架研讨会。

目前,问题框架理论及其扩展等方面有较大发展,例如 Hall 等人提出面向问题的软件工程(Problem-Oriented Software Engineering, POSE)^[3,4],其可应用在安全攸关系统(safety-critical systems)的需求分析和设计^[5,6]、软件系统可靠性关注点分析^[7]等,但是如何把它们应用到实际的软件开发项目中仍是一个有待解决的问题,特别是如何把问题框架的理论研究成果嵌入到实际的需求分析活动中是一个急需解

决的问题。

本原型作品的理论依据是基于以下观察^[8]:软件开发中一个普遍接受的观点是,形式化方法在验证现实世界中的需求问题时遇到困难,而非形式化的方法在识别和表征需求问题中的形式化结构也同样面临困难。计算机处理和解决的问题常处于非形式化的现实世界,而所提供的解决方案本质上是具有形式化属性的机器。这两种截然对立的矛盾在软件开发中总是存在的,因此需要新的方法来划分可形式化部分和无需形式化部分两者之间的界限。我们可以从非形式化的需求及领域描述中发现因果关系,作为问题框架模型(即问题图,下节介绍)形式化变换的前提条件,从而实现用户需求到软件规约的平滑的和可推理的变换。

本原型工具作品是针对以上观察开发出的一种基于问题框架方法的计算机辅助需求工程(Computer-Aided Require-

到稿日期:2013-12-25 返修日期:2014-01-27 本文受国家自然科学基金项目(61262004),广西自然科学基金项目(2012GXNSFCA053010),教育部国际合作与交流司第46批“留学回国人员科研启动基金”,广西科学研究与技术开发计划(桂科合 1347004-22),2013年度大学生创新创业训练计划项目,2013年广西研究生教育创新计划项目(YCSZ2012059)资助。

刘国源(1988-),男,硕士生,主要研究方向为软件需求工程, E-mail: 153123439@qq.com; 万光海(1991-),男,主要研究方向为嵌入式物联网; 庞柳(1988-),女,硕士生,主要研究方向为需求工程; 李智(1969-),男,博士,教授,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)。

ments Engineering, CARE)工具。该工具既实现了问题框架模型的可视化变换又实现了需求文本(英文)的同步修改变换,对需求分析中问题的沟通、建模、分析和软件规格说明的导出具有一定的辅助作用。

2 问题框架建模及变换规则

问题框架方法,是继 Jackson 在软件工程界提出 JSP 和 JSD^[9]方法之后所提出的用于解决软件需求和开发问题的最新方法。该方法强调把关注点放在与计算机软件系统直接交互或间接相关的周围环境中,其原因是基于以下的观察:有软件参与的项目问题的起源往往始于人们观察到某些错误的状态或不应该发生的现象,例如,运行 Windows 98 的 PC 机屏幕出现蓝屏、火车碰撞事故等问题。通过对这些异常状态或不该发生的事件进行一系列分析,往往最终可部分地把问题归结到计算机软、硬件的设计上。除某些科学计算问题(如寻找两个正整数的最大公约数)外,大部分软件系统都不同程度和深度地与现实世界进行交互。因此,广义上的软件开发问题在结构上既应包括计算机中的软硬件又应该包括现实世界中的相关应用领域。解决这样的问题应该充分考虑两者不同的属性、规律和复杂性等。Jackson 把自己 40 多年从事软件开发的经验总结为 5 种基本问题框架,即需求式行为、命令式行为、信息显示、简单工件以及变换,提出并建议解决复杂软件开发问题的有效手段是“分而治之”,即先进行问题分解以得到合适的解决方案,然后尝试进行问题组合并找到最佳解决方案。然而,“分而治之”后得到的子问题往往不能匹配到以上 5 种基本框架,因此需要有一种系统化的方法进一步简化这些子问题,使得需求的描述逐步靠近机器并最终转化为规约机器的描述,这被称为问题变换^[10]。

下面仅介绍与本原型工具相关的问题框架建模与模型变换等内容,对于问题框架其它方面的介绍和阐述,参阅文献[1,2]。问题框架方法建模的主要技术手段是采用问题图来表征计算机软硬件(Jackson 称之为机器)、应用领域及需求三者之间物理上和逻辑上的连接关系,如图 1 所示,它是为糖尿病患者自动注射适量胰岛素控制系统的问题图(该案例的问题描述来源于文献[11],在此给出它的问题图)。

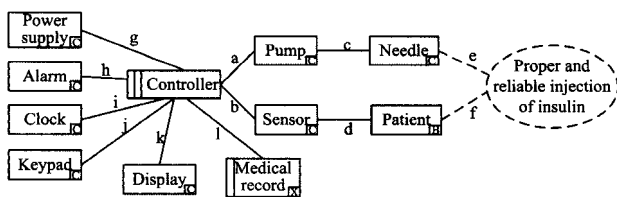


图 1 糖尿病人胰岛素注射控制系统

从图 1 可以看出,问题图至少在以下两个方面对传统的上下文图进行了扩展和补充:

- 用带双竖线的矩形代表包含软件和硬件的机器领域,以示与普通应用领域的区别,如图中的 Controller 领域;
- 普通应用领域用矩形表示,通常是存在于现实世界中的相关物理设备(如图中的 Pump 和 Sensor,右下角标有字母 C——英文 Causal 的简称,表示其行为符合可预测的因果关系),或人(如图中的 Patient,右下角标有字母 B——英文 Bid-dable 的简称,表示其具有一定的主观行为,但还可以顺从外部命令或指导规则)等;

- 带有单竖线的矩形代表可存储信息的领域,以示与普通应用领域的区别,如图中的 Medical record 领域,右下角标有字母 X——英文 Lexical 的简称,表示可存储信息的载体,称为词法领域;

- 用虚线画的椭圆代表需求,需求与所关注的应用领域用虚线连接起来,带箭头的表示约束(必须发生的现象)和不带箭头的表示引用(需求涉及的现象)。

为了实现问题变换,使需求一步一步靠近机器,我们定义以下 3 套变换规则并分别给出实现它们的算法。

因果替换规则:鉴于问题领域的属性主要用来描述该领域和需求之间存在的因果关系,因此可以用来判断是否可以进行等价替换,把需求描述中的“原因”事件替换成领域描述中的“结果”事件,或把“结果”事件替换成“原因”事件。

算法描述如下:

首先把连接需求的问题领域作为起点,如果该问题领域的属性框里面存在形如 $a \rightarrow c, b \rightarrow d$ 这样的关系,而 a, b 属于连接该问题领域的其它领域的事件, c, d 属于连接需求的事件,或者反过来,即说明它们之间存在因果关系,可以进行因果替换。遍历所有连接该问题领域的共享事件,将其保存在一个栈 S 里面,把需求的事件和 S 中的值匹配判断是否形成 $a \rightarrow c$ 或 $c \rightarrow a$ 的关系,如果形成,则把需求的事件替换成 S 中对应的值。

视角转换规则:鉴于连接需求的领域和该领域相邻的领域之间共享的事件相同,则可以把需求连接到它的相邻领域里,因此我们可以从共享事件的“控制者”的角度转换成该事件的“观察者”的角度。

算法描述如下:

从连接有需求的问题领域出发,遍历所有连接该问题领域的其它领域(查找连接该领域的连线的字符串),直到找到一条连线的字符串和该问题领域所连接的需求的连线上的字符串相等时,遍历成功,然后再把连接这个问题领域的需求的连线删除,把需求指向刚刚找到的领域,使需求进一步靠近机器。

删除简化规则:当需求已经完全靠近机器时,该规则对那些需求不再约束或对提及的领域进行移除操作,这时需求只约束或涉及机器领域。至此,原来的问题被转换为一个编程问题,无需考虑任何问题领域。同时在需求描述中加入一些关于这些领域属性的合乎常理的假设。

算法描述如下:

当所有的需求都已经指向机器时,此时需求不在约束或涉及问题领域,应该把所有的问题领域删除,此时遍历所有的问题领域,找到一个便连同它的属性删除,使机器只连接需求。

3 原型开发

为了支持软件开发中需求的获取和分析等活动,设计了计算机辅助软件工程(CASE)工具原型系统。其目的是,能够使用户参与需求的获取、建模分析和变换,并通过工具的使用来验证问题框架建模及变换理论的可行性和实用性。

3.1 设计思想

由于符号本身有令人一眼就能分辨其含义的特点,而且较自然语言更容易描述事物现象之间的关系,容易被人接受,因此画图工具被广泛应用到需求分析中。该工具主要能进行一些简单的图形绘制和文字输入,通过一些具有特殊含义的图形来反映面向问题域的基本思想。同时,该工具还能根据用户的需求,进行需求的变换。变换时主要依据 3 类有科学

依据的规则进行变换,即因果替换、转换视角、删除简化,实现需求的一步一步变换,直到需求得到满足为止。并且该工具还可以对结构化的需求文本进行变换——根据所使用的规则进行相应的文本变换,进而可以让用户体会每一步变换所具有的意义,真正做到从待解决问题域出发,逐步靠近待开发软件系统而最终达到规约其行为的日的。

该工具的主要功能是绘制基于问题框架的一些基本图形、符号及其描述,并表征它们之间在物理上或逻辑上的联系;可以对图形进行添加、修改和删除操作,还可以进行需求变换,同时记录每一步需求文本的变化过程,此外,还可以撤销每一次图形和文本的变换,返回到上一步等。

该工具采用 Microsoft Visual Studio C# 语言开发,PC 机必须安装 Microsoft .NET Framework 4.0 及以上才能正常使用。程序采用模块化设计,同时采用“消息池”技术来消除模块间的引用关系,减少参数集合和返回集合的复杂度。所谓“消息池”就是用来共同存放消息的“池塘”,它不是专属于某个模块,而是为所有模块服务的。采用这个技术是受到单机游戏的存档方式的启发:每次开始游戏时都要去读档,中断游戏前要去存档,下次再读档,游戏就可以继续。发生变化的模块被放到“消息池”,且会被“通知”,并做出相应的动作,这样就可以忽略模块间的依赖关系。

画图工具的主要功能是能够保存和打开预先画好的图形,便于修改。该工具正是基于此原理保存和打开变换前后的图形,通过把所有的实体抽象成字符串,然后保存在相应的文件中,打开时就可以通过字符串恢复原来的实体,实现一一对应的关系,如图 2 所示。

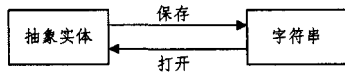


图 2 抽象实体与字符串的一一对应关系

4 案例研究

下面通过糖尿病人注射胰岛素的问题进行案例分析,展示本原型工具是如何工作的。

4.1 问题背景描述

糖尿病是一种体内胰岛素相对或绝对不足或胰岛素本身存在结构上的缺陷而引起的碳水化合物、脂肪和蛋白质代谢紊乱的一种慢性疾病。糖尿病患者经常会并有各种急性病变,急性并发症严重的甚至威胁患者的生命。人体的血糖浓度是波动的,与人的进食、活动、睡眠等有关,而且个体差异很大,胰岛素的生理功能就是稳定血糖,因此人体所需胰岛素的多少是随血糖而变化的。因此研究人员曾提出一种在病人身上安装实时测量血糖浓度的传感器及自动注射人工胰岛素的注射装置,用来模拟胰岛素的生理性分泌模式从而控制高血糖的胰岛素治疗方法。这种装置称为胰岛素泵,是一种医疗电子设备,它关系到患者的生命健康,因此对它的性能要求都非常高,本文利用问题框架的方法来研究如何保证胰岛素泵能更加正常稳定安全可靠地工作;如何在正确的时间给病人注入适当剂量的胰岛素,让它为糖尿病患者更好地服务。

4.2 胰岛素泵的问题图、领域表、现象表

在问题框架中,问题图是对问题领域及用户需求进行建

模的主要手段,它记录并表征了运行软件的计算机、软件应用的上下文、需求及三者关联情况等,胰岛素泵工作原理如下:胰岛素泵是一个仿真胰腺运转的医疗系统。此系统的软件控制部分是一个嵌入式系统,它从传感器(Sensor)收集数据,然后控制泵(Pump)输送指定剂量的胰岛素给患者(Patient),并把注射时间和计量记录在 Medical record 里。它的硬件设备包括控制器(Controller)、注射器(Needle)、传感器(Sensor)、报警器(Alarm)、时钟(Clock)、电源供给(Power supply)、键控(Keypad)、显示器(Display)。要解决的问题是当病人需要输入胰岛素时这个系统能够输送胰岛素;系统必须可靠地运行,并根据当前血糖浓度输入正确剂量的胰岛素,以保持全天血糖稳定,达到控制糖尿病的目的,并在出现异常情况的时候能发出警报给予适当处理。

表 1 问题领域

Name(领域名称)	Description(描述)
Patient(病人)	体内不能正常分泌胰岛素,需要人工注入胰岛素以稳定体内血糖低的一类人
Sensor(传感器)	用于监测感知病人体内血糖浓度的一种装置
Pump(泵)	它用于存储并通过挤压注入胰岛素
Needle(注射器)	它用于将胰岛素注入病人体内
Alarm(报警器)	当胰岛素泵系统出现异常情况时,它发出声音给予用户提醒
Clock(时钟)	它用来为每一时段的胰岛素输注提供严格的时间基准和参照
Power supply(电源供给)	它为整个控制装置提供所需电源
Keypad(键控)	它用于完成胰岛素泵的各种功能的选择、修改、确认
Display(显示器)	它用于通过文字图形和数字等方式显示胰岛素泵的各种信息,让用户实时了解掌握胰岛素泵的工作状态
Medical record(医疗记录)	它用于记录存储过去胰岛素餐前量、输注时间、警告时间、输注总量等

表 2 共享现象

Label(标签)	Name(名称)	Destination(描述)
a	Give(pump_command)	此现象为控制器把注射胰岛素命令发送给泵
b	Receive(concentration_info)	此现象为系统控制器接收传感器发送过来的血糖浓度值并进行计算分析得出应注射的胰岛素量
c	Send(injection_amount)	此现象为泵给注射器发送应注射的胰岛素量
d	Detect(blood_sugar_concentration)	此现象为传感器测量并得出患者体内的血糖浓度值
e	Inject(right_amount)	此现象为注射器直接接触患者并把适量的胰岛素注入体内
f	Detect(blood_sugar_concentration)	此现象为传感器测量并得出患者体内的血糖浓度值
g	Provide(power)	此现象为电源供给设备提供电源
h	Present(alarm)	此现象为遇到异常情况时,控制器让报警器发出警报声给予用户提醒
i	Record(time)	此现象为计时器把当前时间记录下来
j	Set(parameter)	此现象为医生、护士或病人通过键控设备来设定系统工作模式和参数
k	Display(all information)	此现象为显示器显示胰岛素泵的各种信息,为控制胰岛素泵提供良好控制界面和参照
l	Record(injection_info)	此现象为记录胰岛素泵注射的各种信息和过程

我们知道,现实世界中的软件开发问题往往涉及到各种各样的事物,我们分析实际软件开发问题时应根据实际需求

的描述情况选择恰当的抽象粒度进行建模。例如,以上问题中需求的描述大致为传感器测量血液参数,并把参数送到泵控制器,控制器计算血糖浓度,得出胰岛素需要量,然后向一个小型化的泵发送信号使之通过持久连接的针头输送胰岛素,这个过程需要在安全可靠的情况下进行。从上述的问题描述和建模分析中,我们可以确定该问题涉及以下问题领域,如图 1 所示,问题领域的详细描述见表 1,图中的共享现象 a, b, c, d, g, h, i, j, k, l 和需求现象 e, f 等的详细描述见表 2。

4.3 胰岛素泵问题拆分

图 1 是胰岛素泵控制系统所涉及到的整个问题图,它涉及到了许多的问题领域,也是对整个问题边界的宏观描述。我们首先采用“分而治之”的原则对问题进行分解,在这里我们按其功能性需求进行分解,分成图 3 和图 4 所示的两个子问题图,其中子问题 1 完成的主要是胰岛素泵的核心功能性需求,子问题 2 完成的是它的安全可靠性需求。通过分解之后每个子问题各自拥有自己的问题关注点,需求描述就更清晰更有条理了。子问题 1 主要解决如何输注胰岛素才能更好地控制病人的血糖浓度,即对血糖浓度进行测量感知,并把数据报告传给控制器,由控制器分析计算数据并得出胰岛素的注入剂量。子问题 2 主要解决的是胰岛素泵在注射胰岛素过程中所产生的可靠性、安全性问题,比如提示胰岛素泵药池储备已排空,电池电量过低,提示更换电池等,通过分解问题,分解关注点,识别解决问题的困难和注意事项,最终解决问题。

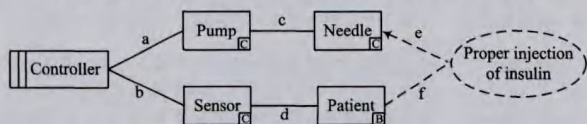


图 3 子问题 1(功能性子问题)

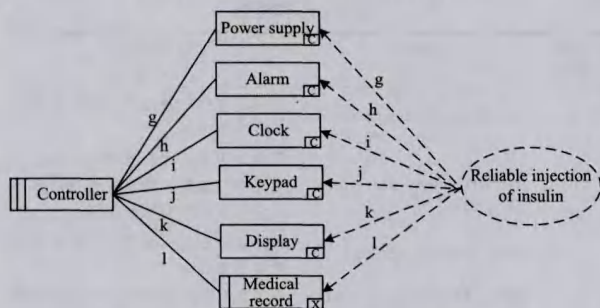


图 4 子问题 2(可靠性子问题)

4.4 案例的问题渐变

根据前面所介绍的问题变换的规则,本节将进行以上子问题 1 的问题变换。

首先根据问题框架方法画出问题图并把各领域之间的共享现象标明,然后给 Sensor、Pump、Needle 这 3 个问题领域添加它们各自存在的因果关系作为领域固有的内在属性,如图 5 所示。

子问题 1 变换第 1 步:我们运用转换视角的变换规则进行变换,由于需求和领域 Needle 之间、领域 Needle 和领域 Pump 之间的共享现象相同,需求和领域 Patient 之间、领域 Patient 和领域 Sensor 之间的共享现象也都相同,因此运用转换视角规则之后,原来指向领域 Needle 和 Patient 的需求重新指向领域 Pump 和 Sensor,这样需求便离机器领域靠近了

一步,如图 6 所示。

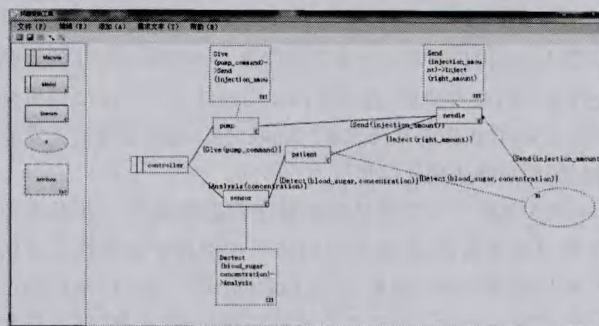


图 5 胰岛素泵子问题 1:问题图

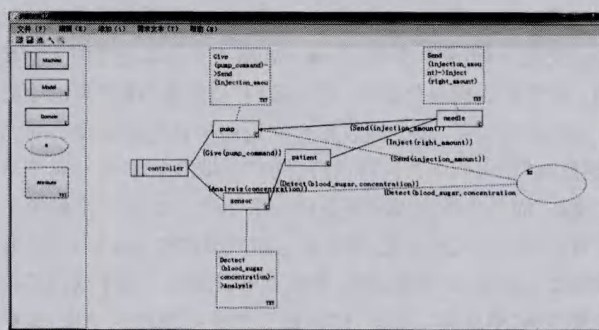


图 6 胰岛素泵子问题 1 渐变步骤 1:运用视角转换规则

子问题 1 变换第 2 步:由于领域 Pump 和领域 Sensor 分别具有以下两个因果关系属性:Give(pump_command) → Send(injection_amount) 和 Detect(blood_sugar_concentration) → Analysis(concentration),因此第 2 步可以运用因果替换规则进行变换,即分别用原因 Give(pump_command)代替结果 Send(injection_amount),用结果 Analysis(concentration)替换原因 Detect(blood_sugar_concentration),如图 7 所示。

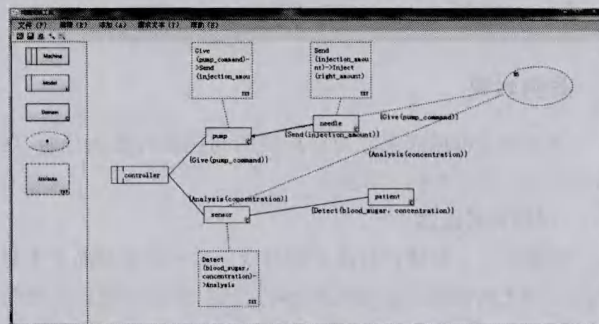


图 7 胰岛素泵子问题 1 渐变步骤 2:运用因果关系转换规则

子问题 1 变换第 3 步,在完成了第 2 步之后,连接需求和领域 Pump 的共享现象与连接领域 Pump 和机器 Controller 的共享现象相同,都为 Give(pump_command),连接需求和领域 Sensor 的共享现象与连接领域 Sensor 和机器 Controller 的共享也相同,都为 Analysis(concentration),因此可以运用视角转换规则进行变换,如图 8 所示,变换之后,需求直接指向机器领域 Controller。

子问题 1 变换第 4 步,在完成了以上 3 步的变换之后,其使得需求只涉及机器领域,运用删除简化规则进行删除简化,如图 9 所示为胰岛素泵问题在进行变换之后的最终的问题图,移除之后需求直接指向机器领域 Controller。

(下转第 168 页)

(EDOC'01), IEEE, 2001;152-161

[12] Yu Y, Huang N, Ye M. Web services interoperability testing based on ontology[C]// The Fifth International Conference on Computer and Information Technology, 2005(CIT 2005). IEEE, 2005;1075-1079

[13] Baldoni M, Baroglio C, Martelli A, et al. A priori conformance verification for guaranteeing interoperability in open environments[C]// Service-Oriented Computing (ICSOC 2006). Springer Berlin Heidelberg, 2006;339-351

[14] Li Xi-tong. A framework for interoperability of BPEL-based workflows[J]. High Technology Letters, 2008, 4;18

[15] 王雷, 徐立臻. BPEL 建模工具中业务流程模型到 BPEL 程序的

转换算法[J]. 计算机与数字工程, 2009, 37(9):20-23

[16] Smythe C. Initial Investigations into Interoperability Testing of Web Services from their Specification Using the Unified Modeling[C]// International Workshop on Web Services-Modeling and Testing (WS-MaTe 2006). 2006;95-119

[17] Hoare. Communicating sequential processes [J]. Communications of the ACM, 1978, 21(8);666-677

[18] Sabnani K K, Lapone A M, Uyar M U. An algorithmic procedure for checking safety properties of protocols [J]. IEEE Transactions on Communications, 1989, 37(9):940-948

[19] 郝瑞兵, 吴建平. 一种形式化的协议互操作性测试方法[J]. 计算机学报, 1997, 20(4):305-359

(上接第 140 页)

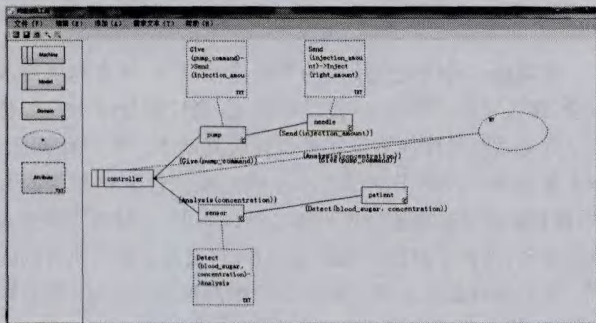


图 8 胰岛素泵子问题 1 渐变步骤 3: 运用转换视角规则

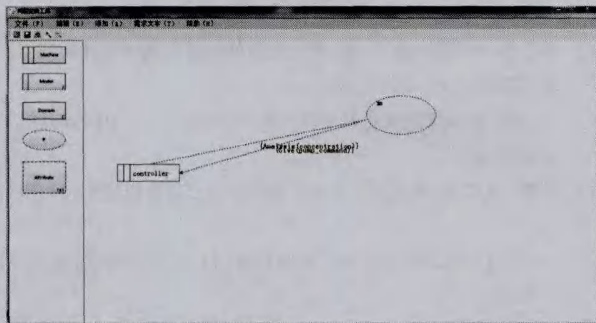


图 9 胰岛素泵子问题 1 渐变步骤 4: 运用删除简化规则

结束语 在本文中我们介绍了一种基于问题框架建模变换的软件需求分析辅助支持工具,它是本文通信作者长期从事问题框架研究的结果^[8,12-14]。目前,该工作的一个早期原型版本可以从网站 <http://www.se.gxnu.edu.cn/tooldemo> 中下载和试用。未来的工作是继续完善本 CASE 工具,并在时机成熟时采用经验软件工程的研究手段对其进行评估和评价。

参 考 文 献

[1] Jackson M. Software requirements and specifications: a lexicon of principles, practices and prejudices [M]. Boston: Addison-Wesley, 1995

[2] Jackson M. Problem frames: analyzing and structuring software development problems[M]. Boston: Addison-Wesley, 2001

[3] Hall G H, Rapanotti L, Jackson M. Problem-oriented software engineering: a design-theoretic framework for software engineering, 2007[C]// Proceedings of the 5th IEEE International Conference on Software Engineering and Formal Methods. Los Alamitos; IEEE CS Press, 2007; 15-24

[4] Hall G H, Rapanotti L, Jackson M. Problem-oriented software engineering: solving the package router control problem [J]. IEEE Transactions on Software Engineering, 2008, 34(2): 226-241

[5] Strunk E A, Knight J C. The essential synthesis of problem frames and assurance cases[J]. Expert Systems, 2008, 25(1): 9-27

[6] Mannering D, Hall J G, Rapanotti L. Towards normal design for safety-critical systems[C]// Fundamental Approaches to Software Engineering. Springer Berlin Heidelberg, 2007; 398-411

[7] Yin B, Jin Z, Li Z. Reliability concerns in the Problem Frames Approach and system reliability enhancement patterns[J]. Jisuanji Xuebao (Chinese Journal of Computers), 2013, 36(1): 74-87

[8] Li Z, Hall J G, Rapanotti L. On the systematic transformation of requirements to specifications [J]. Requirements Engineering, 2013, (doi:10.1007/s00766-013-0173-8), online first article

[9] Berry M D. Software requirements and design; the work of Michael Jackson[J]. ACM SIGSOFT Software Engineering Notes, 2011, 36(2): 39-40

[10] 李智, 金芝. 从用户需求到软件规约: 一种问题变换的方法[J]. 软件学报, 2013, 24(5): 961-976

[11] Sommerville I. Software Engineering 9th Edition [M]. Boston: Addison-Wesley, 2011

[12] Rapanotti L, Hall G J, Li Z. Deriving specifications from requirements through problem reduction [J]. Journal of IEE Proceedings-Software, 2006, 153(5): 183-198

[13] Li Z, Hall G J, Rapanotti L. On the construction of specifications from requirements[C]// Proc of the 14th Workshop on Requirements Engineering. Rio de Janeiro, Brazil: BDBComp, 2011; 431-442

[14] 李智, 庞柳, 刘国源, 等. 一种模型驱动的软件需求分析方法及技术支持[J]. 广西师范大学学报: 自然科学版, 2013, 31(2): 19-26