一种面向同步交互的软件演化过程建模方法

钱 晔[1,3] 李 彤[2,3] 郁 涌[2,3] 孙吉红[4] 于 倩[2,3] 彭 琳[1]

(云南农业大学基础与信息工程学院 昆明 650201)[1] (云南大学软件学院 昆明 650091)[2]

(云南省软件工程重点实验室(云南大学) 昆明 650091)[3] (云南省科学技术院 昆明 650000)[4]

摘 要 全球化软件开发导致以交互方式协作开发的频率和复杂性越来越高。为了控制和规范软件演化的开发行为进而提高软件质量,文献[10]设计了软件演化过程元模型EPMM,由EPMM定义的软件演化过程模型未能形式描述其交互的特点。基于EPMM[10]定义的软件演化过程包括全局层、过程层、活动层和任务层4个抽象层的思想,设计了软件演化过程元模型CEPMM。CEPMM定义的软件演化过程模型可形式描述同步交互的特点是在其活动层,由此提出一种基于通信系统演算(CCS)的软件演化过程活动层的建模方法,然后在Visual Studio平台下实现软件演化过程活动层可视化的建模工具CAmodel。CEPMM构建的软件演化过程活动层模型不仅可以描述并发、迭代等特点,还可以形式描述同步交互的特性,为模型进行严格的数学方法分析、推理奠定了基础。

关键词 软件演化过程的元模型,通信系统演算,同步交互

中图法分类号 TP31 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.8.032

Approach to Modeling Software Evolution Process for Synchronous Interaction

QIAN Ye[1,3] LI Tong[2,3] YU Yong[2,3] SUN Ji-hong[4] YU Qian[2,3] PENG Lin[1]

(College of Basic and Information Engineering, Yunnan Agricultural University, Kunming 650201, China)[1]

(School of Software, Yunnan University, Kunming 650091, China)[2]

(Key Laboratory for Software Engineering of Yunnan Province, Yunnan University, Kunming 650091, China)[3]

(Yunnan Provincial Academy of Science and Technology, Kunming 650000, China)[4]

Abstract In the background of globalization software development, frequency and complexity of interactive collaborative development among software development teams are higher and higher. In order to improve the quality of software by controlling and regulating the behavior of the software evolution development, EPMM was designed in paper [10]. However, the software evolution process model which is defined by the EPMM fails to formally describe the characteristics of synchronous interaction. In this paper, based on four levels(global, process, activity and task) in the software evolution process defined by EPMM, CEPMM was designed. Because it is in activity level that software evolution process model which is defined by CEPMM can describe synchronous interaction of it, an approach to modeling software evolution process in activity level was put forward based on CCS. At last, the activity modeling visualization tool CAmodel of software evolution process was built in visual studio platform. Not only concurrency, iteration and so on, but also synchronous interaction of the software evolution process can be described by model defined by CEPMM, which lay the foundation for analyzing and reasoning mathematically.

Keywords Software evolution process meta-model(CEPMM), Calculus of communication systems(CCS), Synchronous interaction

1 概述

随着计算环境从单机环境向网格计算、分布式计算、云计算以及移动计算等复杂多变的网络环境发展,软件演化的复杂性和普及性越来越高。近年来,软件演化已经成为软件生命周期中最重要的形态之一,成为当今软件工程研究的热点领域[1]。其核心问题是软件如何随着用户需求和外部环境的不断变化而进行灵活的改变。雷曼历时22年总结出的雷曼

到稿日期:2015-07-14 返修日期:2015-10-18 本文受国家自然科学基金项目:软件演化过程的行为验证研究(61262024),国家自然科学基金项目:基于构件的可信软件构造及其行为动态可信测评(61462091),云南省科技厅面上项目:基于构件的可信软件构造及其相关问题研究(2012FB119),云南省教育厅科研重点项目:基于构件的可信软件构造及其可信测评(2013Z057)资助。

钱 晔(1984—),女,博士,讲师,CCF会员,主要研究领域为软件过程方法与技术、计算机应用,E-mail:qy198403@163.com;李 彤(1963—),男,博士,教授,博士生导师,CCF高级会员,主要研究领域为软件过程方法与技术、软件工程;郁 涌(1980—),男,博士,副教授,CCF会员,主要研究领域为软件工程;孙吉红(1983—),男,硕士,CCF会员,主要研究领域为计算机应用;于 倩(1975—),女,博士,讲师,CCF会员,主要研究领域为软件过程方法与技术;彭 琳(1978—),女,博士,副教授,主要研究领域为计算机应用。

定律论证了软件系统必定是不断演化的[2,3]。

通过软件开发实践，人们逐步认识到软件过程可以规范软件开发行为进而提高软件产品的质量[4,5]。软件过程在ISO/IEC 12207被定义为活动的集合，活动视作任务的集合，任务则把输入变为输出[6]。在软件过程领域，Osterweil提出了一个被广泛接受的观点"软件过程也是软件"，认为应像编程软件一样来建模软件过程[7]。软件过程在软件开发中占有至关重要的地位，但是在软件工程领域它是一直未得到适当解决的问题之一，软件演化过程作为控制和规范软件开发行为进而提高软件质量的软件过程之一，是软件演化和软件过程的交叉学科，指软件演化中涉及的一系列过程。它为软件演化构造了一个管理框架，是实施软件演化的工作流程[4,5]。文献[8]认为：软件演化过程可以有效管理软件演化，提高软件演化的效率。

在全球化软件开发的大趋势下，异地和异文化背景下组成的软件开发团队需要冲破地域和文化的障碍[9]，所以协作式开发在软件开发中占据越来越重要的地位，针对这一问题，本研究面向软件演化过程建模，区别于传统的软件过程建模，本文在软件过程建模过程中着重关注软件开发中各开发小组以交互的方式协作开发的特点。

本文基于软件演化过程元模型EPMM[10]定义的软件演化过程包括全局层、过程层、活动层和任务层4个抽象层的思想，设计了软件演化过程元模型CEPMM，CEPMM定义的软件演化过程模型可形式描述同步交互的特点是在其活动层，由此本文提出一种基于通信系统演算CCS的软件演化过程活动层的建模方法，然后在Visual Studio平台下实现软件演化过程活动层可视化的建模工具CAmodel。软件演化过程活动层模型不仅能反映软件演化过程中的并行、迭代等特点，而且相比EPMM建立的模型，能形式描述软件演化过程中同步交互的特点，这与现今全球软件开发的大趋势一致，并且建立的模型是形式化的，为模型使用严格的数学方法分析、推理奠定基础。

2 相关技术

本节主要介绍文中所涉及的EPMM和CCS的主要理论[10]。

2.1 通信系统演算CCS

1980年，Robin Milner在专著《通信系统演算》[11]中提出了CCS (Calculus of Communication Systems)。CCS是一种函数式语言，基本成分是项或称进程，含自由变量的进程称进程表达式，在语法上进程都是由原子操作通过操作符复合而成，操作符的语义都可以通过结构化操作语义定义。进程的组合仍是进程，其组合深度可以是任意的。

定义1[12] 一个进程是一个五元组 $P=(Q,q_0,\Sigma,Act,T)$。其中，$Q=\{q_i\mid 0\leqslant i\leqslant |Q|-1\}$是一个非空有限状态集合；$q_0$是初始态；$\Sigma$是字母表；$Act=L\cup\{\tau\}$是动作集，$\tau$是进程内部动作，$L=A\cup\bar{A}$，$A=\{a\mid a\in\Sigma\}$是接收动作集，$\bar{A}=\{\bar{a}\mid a\in\Sigma\}$是发送动作集；$T\subseteq Q\times Act\times Q$是状态转移关系。

定义2 一个进程P的子进程P'也是进程，子进程同样是一个五元组 $P'=(Q',q_0',\Sigma',Act',T')$。其中，$Q'\subseteq Q$；$q_0'\in Q'$是初始态；$\Sigma'\subseteq\Sigma$是字母表；$Act'\subseteq Act$，$Act'=L'\cup\{\tau\}$是动作集，$\tau$是进程内部动作，$L=A'\cup\bar{A}'$，$A'=\{a\mid a\in\Sigma\}$是接收动作集，$\bar{A}'=\{\bar{a}'\mid a\in\Sigma\}$是发送动作集；$T'\subseteq T$。

定义3[12-14] CCS的进程表达式P可由如下BNF范式表示：

$$P::=0\mid\sum_{i\in I}\alpha_i.P_i\mid P_1\mid P_2\mid \text{new}\ \alpha P\mid A\langle a_1,\cdots,a_n\rangle$$

其中，0表示空进程；$\sum_{i\in I}\alpha_i.P_i$为非确定选择($I$可以是任意一个有穷指标集)，$\alpha_i.P_i$为动作前缀($\alpha_i\in Act$)；$P_1\mid P_2$为并发合成；new αP为限制($\alpha\in L$)；A是进程常量$a_1,\cdots,a_n$是参数。

2.2 软件演化过程元模型EPMM[10]

为关注软件演化过程的不同特性，EPMM[10]定义软件演化过程为包括全局层、过程层、活动层和任务层的一个4层抽象模型，每一层使用不同的形式工具形式描述软件演化过程的不同特点。

EPMM中活动被形式化为一个四元组，即输入输出结构、输出数据结构、局部数据结构和活动体，本文不考虑活动执行所处的资源环境，活动即是活动体。EPMM中一个活动被看作一个活动类，当活动可执行时，创建相应的活动对象，活动对象可以调用其中的任务*Main*，任务*Main*又可调用活动中的其他任务，活动中任务之间可以发生同步交互。

3 软件演化过程的元模型CEPMM

针对EPMM定义的软件演化过程模型不能形式描述其交互的特点，本文以CCS为形式化工具建立软件演化过程的元模型CEPMM(见定义4)，对软件演化过程活动层形式建模，其能形式描述软件演化过程中的同步交互特点。

定义4 在CEPMM中，Ta表示任务集，Ac表示活动集，Pr表示过程集，分别对应于模型中的任务层、活动层和过程层。活动集Ac包含在过程集Pr内，任务集也包含在活动集Ac内。

(1)建立一一对应映射函数 $f_t:Ta\rightarrow Act$，对于$\forall t_a\in Ta$，$\exists a\in Act$，使得$f_t(t_a)=a$，即将每个任务映射为一个动作。

(2)建立一一对应映射函数 $f_a:Ac\rightarrow Q_a$，Q_a为子进程的集合，对于$\forall a_c\in Ac$，$\exists P'\in Q_a$，使得$f_a(a_c)=P'$，即将每个活动映射为一个子进程。活动a_c包含任务集合Ta_1，$Ta_1\subseteq Ta$，根据函数f_t，$\exists Act_1\subseteq Act$，$f_t:Ta_1\rightarrow Act_1$为一个一一对应的映射函数，算子集合$\delta$为$\{.,+,*\}$，动作的子集$Act_1$中的动作通过有限次地使用$\delta$中的算子进行组合而得到子进程，$*$是迭代算子。

(3)建立一一对应映射函数 $f_p:Pr\rightarrow Q_p$，Q_p为进程的集合，对于$\forall p_r\in Pr$，$\exists P\in Q_p$，使得$f_p(p_r)=P$，即将每个过程映射为一个进程。过程p_r包含活动集合Ac_1，$Ac_1\subseteq Ac$，根据函数f_a，$\exists Q_{a1}\subseteq Q_a$，$f_a:Act_1\rightarrow Q_{a1}$为一个一一对应的映射函数，算子集合$\delta$为$\{.,+,*,|\}$，子进程集合的子集$Q_{a1}$中的子进程通过有限次地使用$\delta$中的算子进行组合而得到进程。

4 基于CEPMM活动层建模

在文献[10]中EPMM未对活动形式化建模。本文使用

CEPMM为该活动建立形式模型。首先引入一种非形式化活动建模语言(Activity Modelling Language, AML)规约活动[15],CEPMM以此为媒介建立活动的形式模型。

4.1 活动对象的图形化规约

非形式化语言AML的基本元素分为两类:静态元素和动态元素(见图1),这两类元素的对应关系将在后面进行讨论。

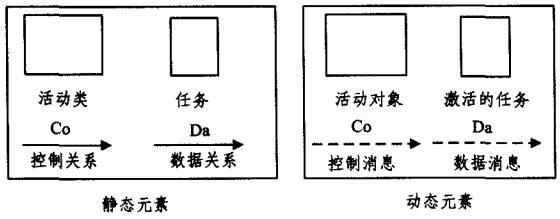

图1 AML的基本元素

文献[10]中,一个活动被看作一个活动类,活动类包含一组任务集,每个任务相当于类中的一个操作,任务与其他实体之间的联系是通过激活的活动类以及内部的任务之间的消息传输来实现的。考虑到任务的原子性,激活的任务执行与消息传递被认为是瞬时的。当活动类被激活时生成活动对象,首先调用它的任务*Main*,激活的任务*Main*又可激活活动中的其他任务。激活的任务可通过消息与其他激活的任务发生同步交互,但这两个任务并不一定属于同一活动对象内部。

任务传输的消息分为两类:控制消息和数据消息。输出方通过控制消息激活输入方;输出方所传递的数据消息是输入方执行所需的。

由消息类型来区分活动与其内部任务*Main*之间以及任务之间具有关系的类型:1)控制关系;2)数据关系(见定义5和定义6)。关系和消息的传递分别为活动内部静态和动态描述。

定义5 在活动类内部,活动与其内部任务*Main*之间以及任务之间具有控制关系,箭头从控制方指向被控制方,表示控制方可以向被控制方发送调用消息,激活被控制方。

若实体双方同时具备控制调用对方的权利,则禁止双方同时调用对方,否则会发生死锁。

定义6 在活动类内部,活动与其内部任务*Main*以及任务之间具有数据关系,箭头表示从控制方指向被控制方,表示控制方可以向被控制方发送数据消息。

若实体双方同时向对方发送数据消息,则认为它们之间发生同步交互,这是本文形式建模唯一考虑到的情况。AML从控制流角度描述活动内部任务之间的结构关系(见定义7),对可能发生同步交互活动的任务采用互补名称命名,为CEPMM形式描述其可同步交互的特点奠定基础。

定义7 活动被定义一个二元组$C_{Ar}=\langle T_a, G_{Ar}\rangle$,$(T_{a1}\cup\{END\})\subseteq T_a$,$T_a\subseteq T_{a1}\cup(\{END, XOR, AND\})$,$T_{a1}$是活动类包含的任务集,$T_{a1}\neq\emptyset$,$T_{a1}$至少包含任务*Main*,$G_{Ar}$是活动以任务*Main*为起点的任务之间的结构关系,任务END标识终止,{XOR,AND}标识活动类所包含的任务之间的基本结构关系,G_{Ar}由基本结构关系嵌套组成,任务之间基本结构关系包括顺序结构、选择结构、并发结构、循环结构,如图2所示。在并发结构中,可能发生同步交互的任务采用互补名称命名。

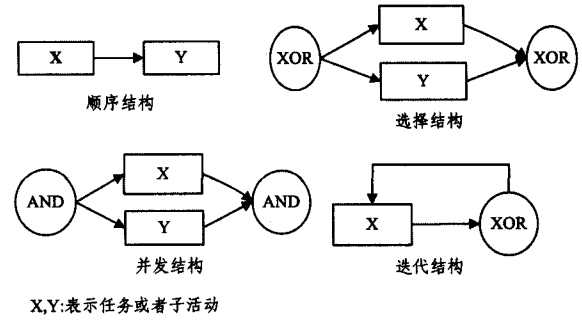

图2 基本结构关系

活动都有相应的子活动,如定义8所示。

定义8 子活动也被定义为一个二元组$C_{Ar}'=\langle T_a', G_{Ar}'\rangle$,$T_{a1}'\subseteq T_{a1}$,$T_{a1}$是活动类的任务集,$T_{a1}'\subseteq T_a'\subseteq(T_{a1}'\cup\{END, XOR, AND\})$,$T_{a1}'$是活动类包含的任务子集,$T_{a1}'\neq\emptyset$,$G_{Ar}'$是它们的结构关系,由基本结构关系嵌套组成。

AML描述基于EPMM建立的软件演化过程模型过程层中的活动a_o的任务结构关系如图3所示,图中b_o是活动a_o的一个子活动,但是d_o不是活动a_o的一个子活动,因为它不包含一个完整的并发结构。

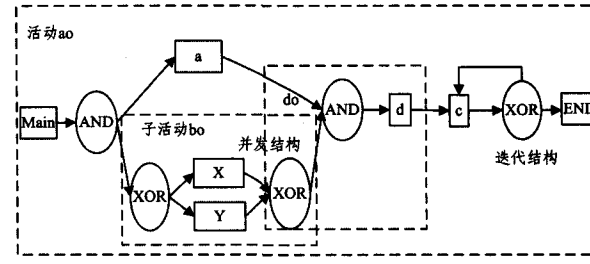

图3 a_o活动AML描述

4.2 AML规约到基于CEPMM模型的转化

AML描述的活动是非形式化的图形化规约,缺乏精确的语义,不便于使用工具对其描述的规约进行动态分析与验证[16]。

为建立以活动的AML规约为媒介的活动的形式化代数形式,AML的活动规约由上述几种基本结构关系组成,CEPMM建立基本结构向基于CCS形式模型转化的规则如图4所示。有限次地使用该准则就能将该活动的AML规约转化为其进程表达式。

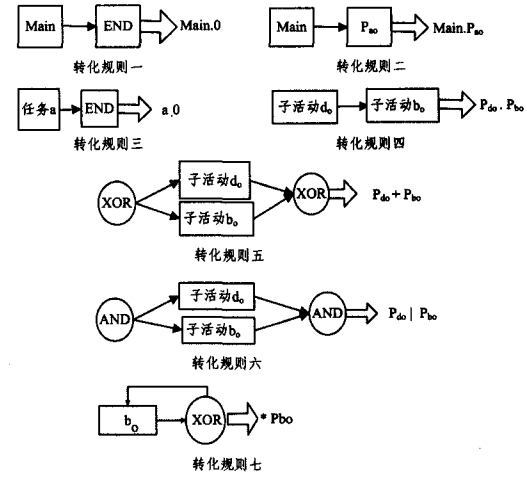

图4 转化规则

CEPMM 对于任意一个活动的 AML 规约可以有限次地使用以上的转化准则，将基于 AML 规约的活动转化为基于 CCS 的形式化模型，此过程即为生成该活动的进程二叉树的过程。

定义 9 活动的进程二叉树叶子结点为任务，即动作，除叶子结点的其他结点都是 CCS 的操作算子，该二叉树的子树为活动中相应的子活动。

每种转化规则都有一种生成活动的进程二叉树方式与其对应，如图 5 所示。CEPMM 借助生成活动的进程二叉树方式建立活动层软件演化过程模型，其是一个自顶向下逐步细化直到生成一棵完整的进程二叉树为止的过程。

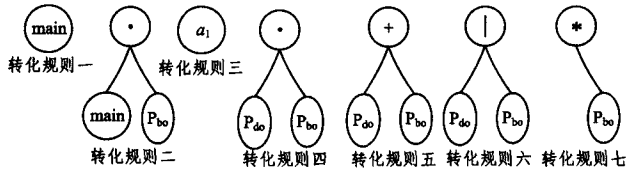

图 5 生成进程二叉树的基本方式

CEPMM 建立活动层软件演化过程形式化模型的主要算法如下。

输入：活动 a_0 的图形化规约

输出：基于 CEPMM 关于活动的代数表达式的前缀表示

```
BEGIN
  初始化进程二叉树 T;
  按照转化规则二的方式生成活动 a0 的进程二叉树 T;
  IF(以 T→rchild 为根结点，进程二叉树对应的活动无任务)
  删除结点 T→rchild;
  Else
    T=T→rchild;
      PbiTree(进程二叉树 T 对应的子活动 AML 规约)
        BEGIN
          IF(子活动只有一个任务)
            BEGIN
              IF(顺序结构)
                按照转化规则三的方式生成子活动的进程二叉树;
                IF(迭代结构)
                按照转化规则三的方式生成子活动的进程二叉树;
        Else
        IF(顺序结构)
          按照转化规则四的方式生成子活动的进程二叉树;
          PbiTree(进程二叉树 T→lchild 对应的子活动 AML 规约);
          PbiTree(进程二叉树 T→rchild 对应的子活动 AML 规约);
          IF(选择结构)
            按照转化规则五的方式生成子活动的进程二叉树;
            PbiTree(进程二叉树 T→lchild 对应的子活动 AML 规约);
            PbiTree(进程二叉树 T→rchild 对应的子活动 AML 规约);
            IF(并发结构)
              按照转化规则六的方式生成子活动的进程二叉树;
              PbiTree(进程二叉树 T→lchild 对应的子活动 AML 规约);
              PbiTree(进程二叉树 T→rchild 对应的子活动 AML 规约);
              IF(迭代结构)
                按照转化规则七的方式生成子活动的进程二叉树;
            END IF
        END PbiTree
  调用二叉树先序遍历算法得序列 s;
END//算法结束
```

5 建模工具 CAmodel 的实现

本文在 Visual Studio 平台下采用 C# 语言开发了一个 CEPMM 活动层软件演化过程可视化的建模工具 CAmodel，主要为用户提供两大功能：1)为活动 AML 规约提供了一个可视化工具；2)将活动的 AML 规约转化为一个基于 CCS 的形式化模型。在建模工具 CAmodel 上构建活动图 3 中 a_0 的活动 AML 规约及其转化后的基于 CCS 的形式化模型 $main.(a \mid (b+c).d).*e$。

结束语 为了控制和规范软件演化的开发行为进而提高软件质量，为了有助于解决全球化软件开发的大趋势下异地异文化的交互问题，本文基于 EPMM 将软件演化过程分为全局层、过程层、活动层和任务层 4 个抽象层的思想，设计了软件演化过程元模型 CEPMM，CEPMM 定义的软件演化过程模型可形式描述同步交互的特点是在其活动层，由此本文提出一种基于 CCS 的软件演化过程活动层的建模方法，然后在 Visual Studio 平台上实现软件演化过程活动层可视化的建模工具 CAmodel。CEPMM 构建的软件演化过程活动层模型可形式描述同步交互的特性，为模型进行严格的数学方法分析、推理奠定基础，但本文未涉足软件演化过程中除同步交互以外其他交互特性的研究以及模型的形式验证，这是下一步研究工作。

参 考 文 献

[1] Dai Fei, Li Tong, Xie Zhong-wen, et al. Towards an algebraic semantics of software evolution process models[J]. Journal of Software, 2012, 23(4): 846-863(in Chinese)
代飞，李彤，谢仲文，等. 一种软件演化过程模型的代数语义[J]. 软件学报，2012，23(4)：846-863

[2] Lehman MM. Laws of software evolution revisited[C]// Proceedings of the 5th European Workshop on Software Process Technology. London, UK: Springer-Verlag, 1997: 108-124

[3] Xie Zhong-wen, Li Tong, Dai Fei, et al. An Approach to Modeling and Normalizing Dynamic-Evolution-Oriented Software Requirements[J]. Journal of Frontiers of Computer Science and Technology, 2012, 6(6): 557-576(in Chinese)
谢仲文，李彤，代飞，等. 面向软件动态演化的需求建模及其模型规范化[J]. 计算机科学与探索，2012，6(6)：557-576

[4] Dai Fei, Li Tong, Xie Zhong-wen, et al. Research on Property Soundness of Software Process Based on EPMM[J]. Computer Engineering, 2014, 40(1): 72-77(in Chinese)
代飞，李彤，谢仲文，等. 基于 EPMM 的软件过程性质合理性研究[J]. 计算机工程，2014，40(1)：72-77

[5] Dai Fei, Li Tong, Xie Zhong-wen, et al. Research on Structure Soundness of Software Processes Based on EPMM[J]. Computer Science, 2013, 40(8): 186-190(in Chinese)
代飞,李彤,谢仲文,等. 基于 EPMM 的软件过程结构合理性研究[J]. 计算机科学,2013,40(8):186-190

[6] ISO, IEC. ISO/IEC 12207: Standard for Information Technology-software Life Cycle Processes[S]. 1998

[7] Osterweil L J. Software Processes are Software Tool[C]// Proc. of the 9th International Conference on Software Engineering. Monterey. USA: ACM Press, 1987: 2-13

[8] Wang Qing, Li Juan. The challenge for software evolution from the Internet[J]. Communications of the CCF, 2009, 5(12): 27-37 (in Chinese)
王青,李娟. 互联网对软件演化的挑战[J]. 中国计算机学会通讯,2009,5(12):27-37

[9] Herbsleb J D, Moitra D. Guest Editors' Introduction: Global Software Development[J]. IEEE Software, 2001, 18(2): 16-20

[10] Li Tong. An approach to modelling software evolution processes [M]. Berlin: Springer-Verlag, 2008

[11] Milner R. A Calculus of Communicating Systems[M]. Lecture Notes in Computer Science, Springer-Verlag, 1980

[12] Milner R. 通信与移动系统 π 演算[M]. 北京:清华大学出版社, 2009

[13] Xiao Fang-xiong, Li Yan, Huang Zhi-qiu, et al. Modeling and Analyzing Web Services Composition Using Timed Probabilistic Priced Process Algebra [J]. Chinese Journal of Computers, 2012, 1(5): 918-936(in Chinese)
肖芳雄,李燕,黄志球,等. 基于时间概率代价进程代数的 Web 服务组合建模和分析[J]. 计算机学报,2012,1(5):918-936

[14] Xiao Fang-xiong, Huang Zhi-qiu, Cao Zi-ning, et al. Process Algebra Extended with Price Information[J]. Journal of Nanjing University of Aeronautics and Astronautics, 2009, 41(1): 69-74 (in Chinese)
肖芳雄,黄志球,曹子宁,等. 一种扩展了价格信息的进程代数[J]. 南京航空大学学报,2009,41(1):69-74

[15] Qian Ye. An Approach to Modelling, Properties Verification and Performance Analysis of Software Evolution Process[D]. Kunming: Yunnan University, 2014(in Chinese)
钱晔. 一种软件演化过程建模、性质验证及性能分析方法[D]. 昆明:云南大学,2014

[16] Wu Shuai. The Research on Translating UML Diagram to B-Method Formal specification and 1st Application[D]. Nanchang: Jiangxi Normal University, 2007(in Chinese)
吴帅. UML 模型图到 B 方法形式规约的转换研究与应用[D]. 南昌:江西师范大学,2007

(上接第 130 页)

参 考 文 献

[1] Ding Zhen-hua, Li Jin-tao, Feng Bo. Research on hash-based RFID security authentication protocol [J]. Journal of computer Research and Development, 2009, 46(4): 583-592(in Chinese)
丁振华,李锦涛,冯波. 基于 Hash 函数的 RFID 安全认证协议研究[J]. 计算机研究与发展,2009,46(4):583-592

[2] Ma Chang-she. Low cost RFID authentication protocol with forward privacy [J]. Chinese Journal of Comptuers, 2011, 34(8): 1388-1398(in Chinese)
马昌社. 前向隐私安全的低成本 RFID 认证协议[J]. 计算机学报,2011,34(8):1388-1398

[3] International Telecommunication Union. ITU Internet Reports 2005: The Internet of Things [R]. Geneva: ITU, 2005

[4] Zhou Yong-bin, Feng Deng-guo. Design and analysis of cryptographic protocols RFID [J]. Chinese Journal of Computers, 2006, 29(4): 581-590(in Chinese)
周永彬,冯登国. RFID 安全协议的设计与分析[J]. 计算机学报,2006,29(4):581-589

[5] Jin Yong-ming, Wu Qi-ying, Shi Zhi-qiang, et al. RFID Lightweight Authentication Protocol Based on PRF [J]. Journal of Computer Research and Development, 2014, 51(7): 1506-1514 (in Chinese)
金永明,吴棋滢,石志强等. 基于 PRF 的 RFID 轻量级认证协议研究[J]. 计算机研究与发展,2014,51(7):1506-1514

[6] Ohkubo M, Suzuki K, Kinoshta S. Hash-chain based for-ward-secure privacy protection scheme for low-cost RFID[C]// Proceedings of the 2004 Symposium on Cryptography and Information Security. Berlin: Springer-Verlag, 2004: 719-724

[7] Miyaji A, Rahman M S. KIMAP: Key-insulated mutual authentication protocol for RFID [J]. Int Journal of Automated Identification Technology, 2011, 3(2): 61-74

[8] Alomair B, Cuellar J, Poovendran R. Scalable RFID systems: A privacy-preserving protocol with constant time identification [J]. IEEE Trans on Parallel and Distributed Systems, 2012, 23 (8): 1-10

[9] Godor G, Imre S. Hash-based mutual authentication protocol for low-cost RFID systems[C]// Proc of the 18th EUNICE Conf on Information and Communications Technologies. Berlin: Springer, 2012: 76-87

[10] Mamun M S I, Miyaji A, Rahman M S. A secure and private RFID authentication protocol under SLPN problem[C]// Proc of the 6th Int Conf on Network and System Security. Berlin: Springer, 2012: 476-489

[11] Wang Shao-hui, Liu Su-juan, Chen Dan-wei. Scalable RFID Mutual Authentication Protocol with Backward Privacy [J]. Journal of computer Research and Development, 2013, 50(6): 1276-1284(in Chinese)
王少辉,刘素娟,陈丹伟. 满足后向隐私的可扩展 RFID 双向认证方案[J]. 计算机研究与发展,2013,50(6):1276-1284

[12] Shamir A. SQUASH-A new MAC with provable security properties for highly constrained devices such as RFID tags [C]// Proc of Fast Software Encryption. Berlin: Springer, 2008: 144-157

[13] Gosset F, Standaert F X, Quisquater J J. FPGA implementation of SQUASH[C]// Proc of the 29th Sysp on Information Theory in the Benelux. Leuven: Werkgemeenschap Informatie, 2008: 1-8