

# Agent 系统软件体系结构形式化建模方法

郑志, 杨德礼, 杨红

(大连理工大学系统工程研究所, 大连 116023)

**摘要:** 基于 Agent 技术为复杂分布式问题提供了求解方法。软件体系结构是控制软件复杂性、提高软件系统质量、支持软件开发和复用的重要手段之一。软件体系结构设计可用于描述 Agent 与 Agent 之间的交互和组织结构的规划, 因此 Agent 系统能从良好的体系结构设计中受益。该文整合了图表句法理论和层次谓词变迁网理论, 提出一种形式化建模方法, 从抽象层(架构)和实现层(动态行为)两方面来构建 Agent 系统的软件体系结构。模型具有可验证和追踪性, 为 Agent 系统软件体系结构分析与评估提供了良好的基础。

**关键词:** 层次谓词变迁网; 图表句法理论; 软件体系结构; Agent 系统; 形式化规约

## Formal Modeling Approach of Agent Systems Software Architecture

ZHENG Zhi, YANG De-li, YANG Hong

(System Engineering Research Institute, Dalian University of Technology, Dalian 116023)

**【Abstract】** The Agent-based technology becomes a method to solve complex and distributed problem. Software architecture is an important means to control the complexity of software systems, to improve software quality and to support software development and software reuse. Agent systems can be benefited from architectural design, which can be used as a plan to show Agents, interactions, and their organizational relationships. This paper represents a formal framework that can be used to specify the architecture of Agent systems by integrating the strength of two notations in a way that the description of a system at the higher level of abstraction defined in Diagrammatic Syntactic Theory(DST) with the concrete realization of it is represented in Hierarchical Predicate Transition Nets(HPrTNs) at a lower level of design.

**【Key words】** Hierarchical Predicate Transition Nets(HPrTNs); Diagrammatic Syntactic Theory(DST); software architecture; Agent systems; formal specifications

### 1 概述

设计和构建高质量的基于 Agent 的系统是一个富于挑战性的任务, 选择合适的体系结构建模方法对 Agent 系统质量属性分析和评估具有重要作用。其中, 可视化和形式化的体系结构建模方法是解决系统中不同利益所有者间交互及防止软件规约不规范等问题的良好方法。当前, 对 Agent 系统及其体系结构的可视化和形式化研究是一个活跃领域。其中一种是整合 Cordy 和 Dean 提出的图表句法理论和高级 Petri-Net 的一种层次谓词变迁网(Hierarchical Predicate Transition Nets, HPrTNs)。前者是基于句法理论并没有形式化语义。句法理论抽象出操作细节, 提供了一个黑盒的系统视角; 仅仅表示做了什么, 并没有表示如何做。句法理论缺乏潜在的语义理论基础, 架构只能用于说明系统的结构(静态的视角), 并不能被形式化分析和评估。而后者是形式化的操作模型说明, 具有良好的分析能力。它可用于推理开发结果的起始点, 以详细的视角(白盒)来建模系统。整合这 2 种方法, 可以互补优缺点对 Agent 系统的体系结构建模, 方法的技术路线是系统的高层次抽象层采用句法理论(便于交互), 在设计低层次具体实现层采用 HPrTNs 表示(便于形式化推理)。

### 2 相关工作

基于 Agent 软件工程的快速发展加快了软件 Agent 的构造过程。在这个范型中, 应用程序被写成 Agent 的集合, 按照一定的通信协议 Agent 与其他 Agent 交互, 例如 Agent 通信语言(ACLs)。Neches 提出的 ACLs<sup>[1]</sup> 依赖一个知识共享问题的三层概念分解, 已应用于知识查询操作语言(KQML)开发中。一些

研究工作也使用了 Petri nets 建模个体 Agent 行为或由 Agent 构成的整个系统, 其表示具有强大完备的数学基础和分析方法, 被广泛地应用在具有并发、并行、异步和随机性质的系统建模与分析中。

在软件工程领域中, 体系结构描述语言(ADLs)加快了大型系统的开发。ADLs 基本的建造块有: (1) 构件(计算和数据存储单元); (2) 连接件(通信协议); (3) 架构配置(构件和连接件的连接图)。设计模型表示法中最重要的考虑是语义架构, 因为它提供了坚实的形式化基础用于描述系统中主要质量属性, 例如分布性、并发性和动态性等。Aesop, Darwin, Wright 和 C2 是近来这方面重要的成果<sup>[2]</sup>。但是大型分布式软件系统包含大量并发运行的构件, 构件之间存在复杂的相互通信与制约关系。已有 ADLs 在描述软件构架动态行为时表达不直观, 缺少可视化、自动化的软件构架动态行为分析验证工具, 使得设计人员难以在系统设计的早期阶段分析及测试系统的正确性、适应性。

### 3 HPrTNs 和图表句法理论

#### 3.1 HPrTNs

HPrTNs 具有定义良好的语义基础, 可应用于描述一个系统的结构(静态语义)和行为(动态语义)<sup>[3]</sup>。

**定义 1** 层次谓词变迁网是一个集合  $HPrTNs=(P, T, F, \rho, SPEC, INSC)$ 。其中,  $P$  是谓词的有限集合;  $T$  是变迁的有限集

**作者简介:** 郑志(1973-), 男, 博士研究生, 主研方向: 信息系统软件开发方法; 杨德礼, 教授、博士生导师; 杨红, 博士研究生

**收稿日期:** 2007-06-08 **E-mail:** dafma@126.com

合； $F$ 是流关系的有限集合，例如  $F \subseteq (P \times T) \cup (T \times P)$ ； $\rho$ 是层次映射，被定义为  $P \cup T \rightarrow \wp(P \cup T)$ ； $SPEC$ 是代数说明，定义为集合  $(S, OP, Eq)$ ， $S$ 和 $OP$ 调用符号， $S$ 是一个类型集合， $OP=(Op_{s1}, Op_{s2}, \dots, Op_{sn}, Op_s)$ 是类型 $s1, s2, \dots, sn, s$   $S$ 的操作集合。 $Eq$ 中的 $S$ -equation定义了 $OP$ 操作含义和属性。另外， $SPEC$ 还用于定义 $HPPrTNs$ 中令牌、流注解、前置条件(约束和契约)等。 $INSC$ 是net inscription，定义为集合  $(\Phi, L, R, M_0)$ 。其中， $\Phi$ 是联结每个  $p \in P$ 与 $S$ 上 $sorts$ 子集的 $sort$ 映射； $L$ 是一个联结每个  $f \in F$ 与 $label$ 的 $label$ 映射； $R$ 是联结每个  $t \in T$ 与一阶逻辑公式的 $constraint$ 映射； $M_0$ 称为初始标记，是从谓词集合 $P$ 到 $Tokens$ 集合的映射。

### 3.2 图表句法理论

图表句法理论(Diagrammatic Syntactic Theory, DST)由Cordy和Dean设计<sup>[4]</sup>，它按照架构中的构件、连接件和关系等重要元素来描述系统。系统是一个集合  $(Element, Types, Connection)$ 。其中， $Elements$ 是一个集合  $(N, C, MS, V)$ ； $N$ 是系统中构件的集合； $C$ 是系统中连接件的集合； $MS$ 是消息点的集合； $V$ 是系统中变量的集合。 $Types$ 是一个用于说明系统元素类型的有限集； $Connection$ 是一个集合  $(src, dst, has, msg)$ 。 $src \subseteq N \times C$ 是一个二元关系，用来建模二元连接件的起源； $dst \subseteq N \times C$ 是一个二元关系，用来建模二元连接件的目的； $has \subseteq N \times MS$ 是一个二元关系，联结构件和消息点； $msg \subseteq MS \times C$ 是一个二元关系，联结消息点和消息连接件。

## 4 Agent 系统体系结构设计的形式化方法

软件 Agent 指自主和灵活的实体，为实现它们的设计目标具有控制自身内部和外部的行为。描述基于 Agent 系统的关键构造是：(1)Agents；(2)Agent 的交互；(3)Agent 的组织。

软件体系结构是软件工程研究的领域之一，指出了构件集和通信的配置。

模型关键构造部分包括：(1)构件(独立可计算的元素)；(2)连接件(系统中连接不同构件的元素)；(3)配置(构件和连接件的配置说明)。

在讨论了 Agent 系统和软件体系结构基本属性后，需阐述软件架构的技术是如何应用到 Agent 系统建模中。特别需要说明以下几个问题：(1)关键的架构构造如何映射成 Agent 系统中关键的抽象模型。(2)架构风格如何用来建模 Agent 系统全局的组织关系和相互的依赖关系。

### 4.1 架构建造

Agent 系统能从所采用的软件体系结构中受益。若选择正确，好的软件体系结构对基于 Agent 系统的软件开发具有如下影响：(1)促进不同利益相关者间通信；(2)展示早期设计决策；(3)使用抽象来加强理解系统的能力；(4)结合多层次复用技术；(5)展示系统演化的程度；(6)提供推理和分析系统行为的坚实基础。因此，采用软件体系结构来建模 Agent 系统。

2 个范型之间需建立一定的匹配关系。对大多数部件来说，Agent 系统和软件体系结构具有共同的部分。例如，两者把交互都视为一阶实体，把计算性实体视为提供一些服务给其他实体或期望其他实体提供的可配置软件部分。然而，Agent 和构件之间的主要区别是 Agent 为非确定性实体，即它们有权利接受或拒绝来自其他 Agent 的请求。而构件是确定性实体，它受限于与其他构件的接口或契约关系，必须尊重其他 Agent 的请求。

表 1 使用一个简单的映射关系，展示了 Agent 系统和软件体系结构关键元素之间的匹配关系。

表 1 软件体系结构与 Agent 系统关键元素映射关系

Agent system	Mapping	Software architecture
Agents	↔	Components
Interactions	↔	Connectors
Organizations	↔	Configurations

### 4.2 架构风格

本节讨论如何在高层次设计(抽象架构)和低层次设计(具体架构)中说明 Agent 的结构和行为。同时，利用该方法，个体 Agent 行为和它们的交互也得到研究。

以下案例是一个关于 E-ReviewSystem(ERS)系统的简化说明，这是一个在线评阅和编辑系统，评阅人、编辑以电子文档方式返还评阅和编辑报告。图 1 表示 ERS 流程。系统由评阅人(Reviewer)、编辑(Editor)和关系构成。

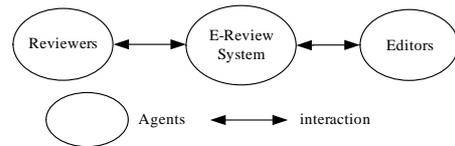


图 1 ERS 系统的工作流程示意图

图 2(a)展示了 DST 中的 Client/Server 架构风格的实例，圆形表示任务；矩形内双箭头表示一个过程调用连接件；“+”操作符表示一个或多个连接件的实例。图 2(b)展示了 ERS 系统的架构，匹配于图 2(a)的 Client/Server 架构实例。通过映射图 1 中 Agent 和交互为对应的构件和连接件，获得了 ERS 架构。连接件用于建模 Reviewer, Editor, E-Reviewer Agent 之间同步的通信。除非具有一个或多个责任安排的责任安排，当前的工作做完，否则不允许它们请求更多的评阅或编辑工作。

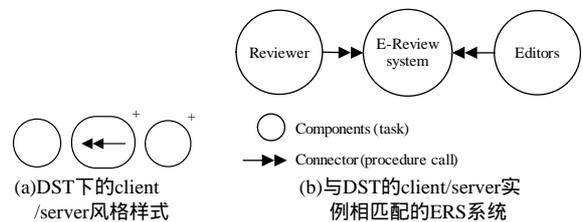


图 2 基于同步交互方式的 DST 中的 ERS 抽象句法架构

图 3 展示了 HPPrTNs 下 ERS 表示的抽象语义，可通过以下步骤获取：(1)映射 tasks(圆形)为 super transitions(点状矩形)；(2)映射 connectors(双箭头)为 super predicates(点状圆形)；(3)映射  $dst, src$  二元关系为 HPPrTNs 中的 flows；(4)使用基于模式的精化技术来精化 HPPrTNs 中的元素<sup>[5]</sup>；(5)复查规约说明，查找目标系统中是否所有应用需求都被满足。此步可能回退以前步骤来确认规约正确性、完整性、一致性。

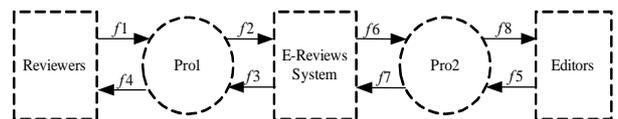


图 3 HPPrTNs 的 ERS 抽象语义架构

图 4 展示图 3 中 HPPrTN 模式实例子网中超级节点的精化。图 4(a)展示了 Editor(Reviewer)的精化，此为 Loop 模式的实例。图 4(b)展示了连接件的精化，它基于 procedure call connector 的语义，用于建模阻止 Reviewer 和 Editor 已经有 review 和 editing 安排的情况。图 4(c)展示了 E-review Agent 的精化，此为并行路由模式的实例；说明一个多重冲突树工作流，它可按任意顺序执行。

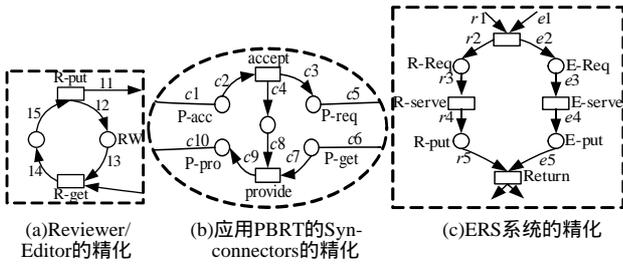


图 4 应用 PBRT 各节点的精化

图 5(a)和图 5(b)展示在假设评阅文章数量没有限制情况下, DST 模式的异步通信情况的 ERS 系统。异步通信方式中,除了通信部件,图 5(a)和图 5(c)中的大多数部件与图 2 相同。为了建模异步通信(没有死锁),通信部件可通过矩形中的点状箭头(invocation connectors)来说明。图 5(c)展示了使用 PBRT, invocation 方法的精化过程。connector 表示通信模型,模型内一个过程诱发了另一个过程。

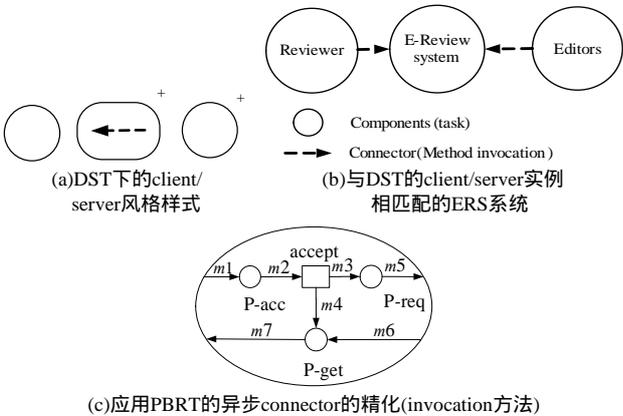


图 5 异步交互方式的下的 ERS 抽象句法架构与 connector 的精化

(上接第 34 页)

要说明的是,该实验中由于排序组件的替换不涉及组件间状态的迁移,本质上属于无状态组件的动态替换,其程序类关系如图 4 所示。

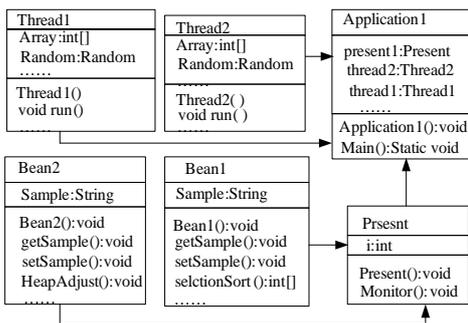


图 4 程序类关系

实验结果表明,当互斥运行的线程发生切换,即系统运行环境发生变化时,运行于监测代理内的基础监测代码能自动监测到这些变化,并根据所生成的数组长度的不同动态替换进所需的排序组件。对该数组进行排序,实现了动态适应系统环境变化和系统可用性的目标,验证了在关键任务系统中实现任务热插拔的可行性,达到了预期的实验效果。

#### 4 结束语

任务热插拔为保证关键任务系统的可用性探索了一条新途径,而任务热插拔的实现又是一个难点问题。本文在分析

#### 5 结束语

本文讨论了一个整合的形式化方法,用于建造 Agent 系统的软件体系结构。该方法中的模型通过提供一种语境,例如活锁、死锁、安全性等系统属性能够得到研究,从而有助于开发更健壮、可靠、灵活的 Agent 系统。

同时,本文还提出了一个 Agent 系统建模的全过程。开始于非形式化的系统说明,终止于具有隐含语义理论的系统规约,详细说明了操作行为和 Agent 系统间的通信。

本文并未讨论 Agent 中与 AI 相关的问题,例如信念、意图、渴望、知识、know-how 等意图属性。这些被认为是 Agent 设计和理论研究中最重要的课题。为了表现这样的能力,还需要扩展相应的模型。一种可能的方法是结合面向对象技术共同建造 Agent 行为能力,以后将进行更深入的工作,讨论这些课题。

#### 参考文献

- [1] Neches R, Fikes R, Finin T, et al. Enabling Technology for Knowledge Sharing[J]. AI Magazine, 1991, 12(3): 36-56.
- [2] Medvidovic N, Taylor R. A Classification and Comparison Framework for Software Architecture Description Languages[J]. IEEE Trans. on Software Engineering, 2000, 26(1): 70-93.
- [3] He Xudong. A Formal Definition of Hierarchical Predicate Transition Nets[C]//Proc. of the 17th International Conference on the Application and the Theory of Petri Nets. Osaka, Japan: [s. n.], 1996.
- [4] Cordy J, Dean T. A Syntactic Theory of Software Architecture[J]. IEEE Trans. on Software. Eng., 1995, 21(4): 302-313.
- [5] Reza H. An Integrated Method to Software Architecture[C]//Proc. of SEKE'03. San Francisco, USA: [s. n.], 2003.

国内外相关实现技术的基础上,重点探讨了关键任务系统中任务热插拔的实现技术,经初步的实验验证,达到了预期的效果。在对无状态任务组件的热插拔研究的基础上,关键任务系统中有状态任务组件的热插拔将是下一步研究的重点。

#### 参考文献

- [1] Plasil F, Balek D, Janecek R. SOFA/DCUP: Architecture for Component Trading and Dynamic Updating[C]//Proc. of the International Conference on Configurable Distributed Systems. Annapolis, Maryland, USA: [s. n.], 1998: 35-42.
- [2] Glass G, Cao Pei. Adaptive Page Replacement Based on Memory Reference Behavior[C]//Proc. of the International Conference on Measurement and Modeling of Computer Systems. Seattle, USA: ACM Press, 1997: 115-126.
- [3] Fox A, Kiciman E, Patterson D A. Combining Statistical Monitoring and Predictable Recovery for Self-management[C]//Proc. of Workshop on Self-healing Systems. Newport Beach, CA, USA: [s. n.], 2004: 49-53.
- [4] Shen Junrong, Sun Xi, Huang Gang, et al. Towards a Unified Formal Model for Supporting Mechanisms of Dynamic Component Update[C]//Proceedings of Symposium on Foundations of Software Engineering. New York, USA: ACM Press, 2005: 80-89.
- [5] Soules C A N, Appavoo J, Hui K, et al. System Support for Online Reconfiguration[C]//Proceedings of USENI X Annual Technical Conference. San Antonio, Texas, USA: [s. n.], 2003: 141-154.