

文章编号:1000-582X(2002)10-0132-04

# 面向 Agent 的软件工程\*

郭亮, 黄席樾

(重庆大学自动化学院, 重庆 400044)

**摘要:**面向 Agent 的软件工程是当今软件工程中新的研究热点。相对于原有的开发方式,在现今软件系统越来越复杂的开发活动中,特别是在用 Agent 表示高层次抽象行为实体的软件系统中,通过组织 Agent 成为具有独立能力和协作的部件,形成比已有的软件方法更加适合的开发方法。在开发基于 Agent 系统中采用高层次、特殊软件工程方法研究的基础上,讨论并分析了这些方法的特点,以及软件工程中相关技术针对 Agent 实现的扩充,为实际建立基于 Agent 的软件系统提供了参考。

**关键词:**智能体;软件工程;统一建模语言;设计模式;构件

**中图分类号:**TP311.5

**文献标识码:**A

面向 Agent 的软件工程作为软件工程新的研究范例在文献[1]中进行了描述。但是,为了成为软件业中的新范例,必须发展鲁棒、易用的方法和工具。

首先解释什么是 Agent。Agent 也叫作软件 Agent、智能 Agent,它是一个自治的软件体。其智能性体现为这种软件有一定的基于知识的智能行为。

通常,用强/弱定义来分类 Agent<sup>[2]</sup>。弱概念的 Agent 有自己的意念(自主性),它们能够进行相互交流(社会性),它们能对外界刺激产生反应(反映性),能采取主动行动,表现出面向目标的行为(主动性)。强概念的定义中,Agent 除了具有弱概念的特性外还具有通常人类具有的部分特征,如:目的,信念,义务,移动性等等。

## 1 基于 Agent 软件工程中的术语

面向基于 Agent 的编程(AOP)是面向对象编程(OOP)的扩展。它涉及到编程语言和实现水平。AOP 由 Shoham 在 1993 提出<sup>[3]</sup>。

面向 Agent 的开发(AOD)是面向对象开发(OOD)的扩展,涉及到整个开发流程,除了编程外,还包括需求,分析,设计。

基于 Agent 的软件工程,多 Agent 的系统工程(MaSE),以及面向 Agent 的软件工程(AOSE)实际上具有相同的意思。AOSE 与 AOD 的不同在于前者除了开发系统外,还包括重用和维护多 Agent 系统。

基于 Agent 的计算可以用于描述所有与 Agent 相关的软件工程中。它包括采用什么样的 Agent 和 Agent 间如何进行计算。

## 2 面向 Agent 的软件工程

面向 Agent 的软件工程是用比较便宜的方法和工具来开发和维护基于 Agent 的软件系统,让其灵活、易用、质量高并具有可升级性,等同于同面向对象的软件工程在软件开发中的作用。

### 2.1 Agent 和对象的区别

面向 Agent 的编程(AOP)可以看作面向对象编程(OOP)的扩充。OOP 是结构化编程的延续,它的主要实体是其中的对象。对象是包含了数据和方法的逻辑单元,它成功地用于描述现实世界的被动实体(如房子)。而 Agent 有可能是对象的继承者,它能够改良对主动实体的描述。Agent 类似于对象,但它的结构还支持反映主动对象的心智模型,如信仰和承诺等。另外,Agent 间通过 Agent 的通讯语言(如,ACL, KQML<sup>[4]</sup>)支持高层次的相互交流。

AOP 与 OOP 的另一个不同是对象由外部进行控制(白盒控制),而具有自治行为的 Agent 不能从外部完全控制(黑盒控制),既 Agent 可以说“不”。

### 2.2 Agent 的能力缺陷

由于 AOSE 是一个新兴的快速成长的研究领域,很多研究学者对 AOSE 的能力显得过分乐观。

\* 收稿日期:2002-06-12

作者简介:郭亮(1974-),男,重庆人,重庆大学硕士研究生。研究方向:人工智能,Agent 技术。

Wooldridge 和 Jennings<sup>[5]</sup>总结了 AOSE 的 10 多个隐患并分类为 5 个组:策略的、概念的、分析设计的、Agent 层面的和社会层面的缺陷。策略的缺陷可能出于过度吹嘘 Agent 的能力,并想寻找一种通用的解决方法;概念上的缺陷是开发人员忽略了 Agent 是多线程的软件;当开发人员忽略了软件技术后,分析设计上的缺陷有可能出现;Agent 层面缺陷的出现则有可能是开发人员试图用太多或太少的人工智能在基于 Agent 的系统中;最后,社会层面的缺陷可能是由于开发人员在开发中把 Agent 用得太多或太少造成,造成 Agent 的角色、分工协作等不能正确对应实际开发的系统。

### 3 高层次的方法

以下说明在基于 Agent 的系统建模和开发中用到的从上到下和迭代的各种方法。

#### 3.1 Gaia 方法学

Wooldridge, Jennings 和 Kinny<sup>[6]</sup>展示了面向 Agent 的分析和设计的 Gaia 方法学。Gaia 是一个支持 Agent 开发的小粒度 (Agent 的结构) 到大粒度 (Agent 社会结构) 的方法学,不过它要求在运行前就定义好 Agent 间的联系和结构。Gaia 的动机是为了解决现存的方法不能表示自治 Agent。其弊端在于不能很好的对 Agent 间交互的质量和建模。使用 Gaia, 软件开发者能按照系统需求进行系统的开发和有序地实现。

Gaia 中分析过程的第一步是发现系统中的角色。第二步在已发现的角色中对其交互进行建模。角色有 4 个特性:职责,许可,行为,协议。职责分为两种:活性(有利于系统的行为),安全性(防止对系统的破坏)。许可表明哪种角色可以做什么访问何种信息。行为是角色自身的任务。协议是角色间交互的规范和模式。Gaia 中定义有各种角色的模板,其中规范了属性和交互的协议。

Gaia 中设计过程的第一步是映射角色到 Agent 的类型,并产生正确数量的 Agent 实例。第二步是决定满足角色或 Agent 实现的服务模型。最后一步是产生代表 Agent 间通讯的识别模型。

由于提到的局限性,Gaia 在开放的不可预测的 Internet 应用领域中价值不大。但它也被证明在封闭的基于 Agent 的系统中应用良好。Jennings 对此进行了扩展和构想以支持基于 Internet 的应用开发。

#### 3.2 多 Agent 系统工程的方法学

Wood 和 DeLoach<sup>[7]</sup>提出多 Agent 系统工程的方法学 (MaSE)。MaSE 类似于 Gaia, 但 MaSE 还支持用 MaSE 工具进行代码的自动产生。MaSE 的动机在于解

决当前匮乏的建立基于 Agent 系统的方法和工业开发工具,目标是引导设计人员从初始的系统规范到系统实现。领域限制方面, MaSE 和 Gaia 一样,而且它还要追求 Agent 的交互是单对单而不是多点传送。

MaSE 分成 7 个阶段:目的捕获(把最初的系统规范转换为系统目标的层级结构)。首先基于最初的系统需求标识出目标,然后按目标在层级中结构和主题顺序的重要性进行排序。第二阶段,利用用例技术按照最初的系统规范建立用例和序列图。用例表示了各种角色间的逻辑交互路径。序列图表示角色间的最小化信息传送。第三阶段是角色精练。通常,一个角色代表一个目标,几个目标也可映射到一个角色。通过角色建立一组与角色目标相关的任务,这些任务定义为状态图。第四阶段,映射角色到 Agent 类。这些类间的联系相对于层次类图有更高的语意联系。第五阶段,建立 Agent 间的会话,按照表示会话的状态图定义通讯协议。第六阶段,组装 Agent 类,实现 Agent 内部的功能函数。这些功能函数的选择按照 Agent 的分类(信念—愿望—意图型(BDI)、反应型、计划型、基于知识型,用户自定义型)进行。最后阶段,系统设计阶段,按照 Agent 类产生 Agent 实例,并用实施图来表达系统设计。

#### 3.3 数据库信息系统建模的扩展方法

Wagner<sup>[8]</sup>提出基于 Agent 对象联系(AOR)模型化的信息系统设计方法。AOR 的灵感来自于数据库中的实体联系(ER)元模型和关系数据库模型(RDB)。实体联系(ER)元模型能很好的支持静态实体或对象,但不利于对主动实体或 Agent 建模。AOR 模型化通过提供对 Agent 间联系的建模扩展了 ER 建模法。

在 AOR 中,实体分为 6 类:Agent、事件、行为、承诺、要求和对象。“承诺”和“要求”是相对的,一个 Agent 的承诺也就是另一个 Agent 的要求。组织机构由一组子 Agent 系统组成,每一个子系统拥有权利执行一定行动,它们同时也承担一些职责,如对 Agent 组织的要求和相关事件进行监视。

## 4 设计方法

以下是对面向对象软件工程的设计方法和标准在基于 Agent 的软件工程中的扩展。

#### 4.1 UML(Unified Modeling Language)

UML 是一种系统建模语言,它支持软件工程的各个开发周期,支持先进的软件技术并具有广阔的应用领域。Yim<sup>[9]</sup>等对 UML 进行了扩展,产生了对象构建语言(OCL),它支持把面向 Agent 的问题模型转换为面

向对象的问题模型。转换中采用设计模式表达 Agent 间的联系,然后用对象间的关系进行表述。

Odell<sup>[10]</sup>提出一种表示 Agent 交互协议的 3 层表示法(AIP)。AIP 定义信息交流的表示模式,并且对 UML 的包,模版,序列图,协作图,活动图及状态图进行了扩充。第一层用包,模版表示通讯协议,第二层用序列图,协作图,活动图及状态图表示 Agent 间的交互。第三层用活动图及状态图表示 Agent 内部的过程。Odell 后来还针对 Agent 扩展了 UML 形成 AUML,并提交给 UML 标准化组织。

#### 4.2 设计模式

设计模式是指重复出现的程序代码或软件构架中构件的模式。Aridor 和 Lange<sup>[11]</sup>提出一种针对移动代理的设计模式分类方案,并提出属于每个类的相应模式。其用意是在开发基于移动代理的系统开发中提高重用和代码质量。分类为:移动类(表示在不同环境中迁移的模式类型),任务类(表示移动代理的任务处理模式),交互类(表示代理间的通讯和协作模式)。

Kendall<sup>[12]</sup>提出一种针对 Agent 的服务层构架模式,各种模式属于相应层次。7 个服务层分为移动、转换、协作、行动、推理、信念和感觉等模式。这些模式的分类和选择已经由以前的成功实现所证实。这种分层的构架有点类似于模式的逻辑分组。移动层、转换层负责处理类的迁移;协作层负责类的交互。行动层负责类的任务实施。

以上两种方法在移动代理研究中的不同是后者包含了所有主要的 Agent 设计模式。

#### 4.3 软件构件

软件构件是一组逻辑上相关联的具有一定的功能的模块,它为对象提供了活动的空间。通过组合相关对象,构件形成粒度更大的、可重用的代码组,这有助于对系统进行分析和设计。现在,基于构件的软件工程是一个有效和流行的开发方式。然而,相对于 Agent 而言,构件不是一个自治体。

Erol, Lang 和 Levy<sup>[13]</sup>提出一种 3 层构架的应用构件的 Agent 系统开发方式。第一层是交互层,由 Agent 角色和内容建立。第二层是能存储信息(如计划、执行状态等)的局部信息和专门技能。第三层是信息内容,表示特定领域的知识等,如打包的遗留系统。

### 5 结束语

对当前基于 Agent 系统的研究方法进行概述,关注的焦点主要在通用的高层次的方法和关联到软件工

程的特例方法。范围没有包括特别的 Agent 方法,如提高协同、协作、通讯、人工智能等。今后的工作包括对这一领域更加彻底地分析和对这些方法的测试和实践,以更好地联系 Agent 理论与相应软件工程的实现。

#### 参考文献:

- [1] LIND J. Issues in Agent - Oriented Software Engineering[EB/OL]. <http://citeseer.nj.nec.com/lind01issues.html>, 2000 - 01/2002 - 01.
- [2] WOOLDRIDGE M J, JENNINGS N R. Intelligent Agents: Theory and Practice[J]. Knowledge Engineering, 1995, 10(2): 115 - 152.
- [3] SHOHAM Y. Agent - oriented programming[J]. Artificial Intelligence, 1993, 60: 51 - 92.
- [4] LABROU Y, FININ T, PENG Y. Agent Communication Languages: The Current Landscape[EB/OL]. <http://www.cs.umbc.edu/~jklabrou/publications/ieeeIntelligentSystems1999.pdf>, 1999 - 03/2002 - 01.
- [5] WOOLDRIDGE M J, JENNINGS N R. Pitfalls of agent - oriented development [EB/OL]. <http://agents.umbc.edu/introduction/paod.pdf>, 1998 - 09 - 13/2002 - 02.
- [6] WOOLDRIDGE M J, JENNINGS N R, KINNY D. A methodology for agent - oriented analysis and design[EB/OL]. <http://www.csc.liv.ac.uk/~mjw/pubs/agents99.pdf>, 1999 - 03 - 03/2002 - 02 - 08.
- [7] DELOACH S A. Multiagent Systems Engineering A Methodology and Language for Designing Agent Systems[EB/OL]. <http://www.cis.ksu.edu/~sdeloach/publications/Conference/MaSE-AOIS99.pdf>, 1999 - 03 - 15/2002 - 03 - 10.
- [8] WAGNER G. Agent - Object - Relationship Modeling[EB/OL]. <http://citeseer.nj.nec.com/wagner00agentobjectrelationship.html>, 2000 - 04 - 05/2002 - 02 - 04.
- [9] YIM H, CHO K, JONGWOO K, et al. Architecture - Centric Object - Oriented Design Method for Multi - Agent Systems[EB/OL]. <http://www.auml.org/auml/working/yim-w001.pdf>, 2000 - 02/2002 - 03.
- [10] ODELL J, PARUNAK H V D, BAUER B. Representing Agent Interaction Protocols in UML[EB/OL]. <http://www.jamesodell.com/Rep-Agent-Protocols.pdf>, 1999 - 04/2002 - 03.
- [11] ARIDOR Y, LANGE D B. Agent Design Patterns: Elements of Agent Application Design[EB/OL]. <http://www.moe-lange.com/danny/patterns.pdf>, 1998 - 07 - 13/2002 - 03/10.
- [12] KENDALL E A, KRISHNA P V M, PATHAK C V, et al. Patterns of intelligent and mobile agents[EB/OL]. <http://www.cs.toronto.edu/km/tropos/patterns-agent98.pdf>, 1998 - 03 - 20/2002 - 03.
- [13] EROL K, LANG J, LEVY R. Designing Agents from Reusable Components[EB/OL]. <http://www.i-a-i.com/Publications/Agents2000DivaPoster.pdf>, 2000 - 06 - 04/2002 - 04 - 03.

## Agent – Oriented Software Engineering

*GUO Liang, HUANG Xi -yue*

(College of Automation, Chongqing University, Chongqing 400044, China)

**Abstract:** Agent – Oriented software engineering is the one of the most recent contributions to the field of software engineering. Agent becomes isolation and negotiation component, compared to existing development approaches, it has advantage in system development, especially where Agent represents high-level abstractions of active entities. Through the research of high-level and specific methodologies in the agent-based software engineering, this paper analyses their characteristics and some extension of software technology for implement Agent, gives some references to build agent-based system.

**Key words:** intelligent agents; software engineering; UML; design patterns; components

(责任编辑 张 苹)

~~~~~  
(上接第 131 页)

## Intrusion Detection Based on Data Mining

*LIU Yong -guo, LI Xue -ming, LIAO Xiao -feng, WU Zhong -fu*

(College of Computer Science and Engineering, Chongqing University, Chongqing 400044, China)

**Abstract:** As computer networks play increasingly vital roles in modern society, information security becomes one of the most important research issues in the field of information technology. But intrusions cause a serious security risk, how to efficiently prevent and detect intrusions becomes one of hot research problems in the field of information supervision. The traditional process of building the model of intrusion detection is slow, whose cost of research and development is high. However, data mining has unique advantages in acquiring unknown knowledge. So, intrusion detection based on data mining becomes a hot issue. The research background, architectures, techniques, problems to be solved and the future direction are discussed after analyzing current status of network intrusion and situation of R&D on intrusion detection and data mining.

**Key words:** intrusion detection; data mining; information security

(责任编辑 吕赛英)