

面向 Agent 软件体系结构的团体特征研究

相洪贵, 李卫华

(广东工业大学 计算机学院, 广东 广州 510090)

摘要: 介绍了软件体系结构的研究现状, 研究了其团体特征, 设计出了一个具有相互学习能力、相互协调和合作的由 Agent 构件组成的团队. 用形式化语言 Object-Z 对该软件体系结构进行了描述, 总结出了构建一个高效率团队的方法及评价其效率的规则.

关键词: Agent; Object-Z; 软件工程; 软件体系结构; 团队

中图分类号: TP311. 5

文献标识码: A

文章编号: 1007-7162(2006)02-0092-06

随着计算机硬件技术的发展和软件复杂性的增加, 在软件设计中, 软件的局部和整体系统结构方面已经变得越来越重要, 软件体系结构概念的提出正说明了软件设计技术逐步走向成熟. 软件体系结构是由结构和功能各异、相互作用的部件集合, 按照层次构成的. 它包含了系统基础构成单元、它们之间的关系、在构成系统时它们的合成方法以及合成约束的描述^[1].

大型软件产业的发展使得软件的体系结构越来越复杂, 个体的有限智力和精力与技术的多学科交叉以及迅猛发展之间存在严重的不平衡. 当今的软件开发中, 由于地域上、时间上的限制, 传统的局限于小范围内的团队合作被打破, 把更多的任务转向使用 Internet 作为一种分布这些任务的基础设施已经成为一种趋势. 组建分布式的团队在大型的软件开发中成为一个关键环节.

Agent 技术就是一种能够适应这种分布式环境的前沿技术. 在软件工程领域, Agent 技术已经引起了人们的重视, 研究人员认为 Agent 技术将成为新一代复杂的、分布式计算机系统设计的软件工程技术^[2].

1 软件体系结构的研究现状

目前, 软件体系结构仍是一个非常新的研究领域, 整个软件体系结构技术从理论到实践还处在发展的阶段, 如: 缺乏统一的概念和坚实的理论基础; 缺乏工程知识的系统化和标准化及形式化; 没有建立统一的体系结构的工程描述方法; 可应用的体系结构的结构工程工具研究仍在试验室阶段等等^[1]. 但软件体系结构作为软件工程领域中的一个组成部分, 已经取得了长足的发展, 现已成为国内外研究的热点. 目前较活跃的研究方向包括^[3]:

- 1) 软件体系结构形式基础的研究.
- 2) 针对软件体系结构描述中特有的问题研究新的高级语言.
- 3) 建立用于度量和评价软件体系结构的模型和方法.
- 4) 建立面向专门领域的软件体系结构范型库.

收稿日期: 2005-02-25

基金项目: 广东省自然科学基金资助(032496)

作者简介: 相洪贵(1989), 男, 在读研究生, 主要研究方向为网络软件.

5) 把软件体系结构从目前的直觉和经验状态过渡到理论.

2 面向 Agent 的软件体系结构

2.1 基于 Agent 的构件

到目前为止, 最广泛被人们接受的 Agent 的概念是在 1995 年 Wooldridge 和 Jennings 提出的: Agent 是在某个环境中的计算机系统, 该系统有能力在这个环境中自主行动以实现其设计目标^[4]. 还有人提出了: Agent 是设计用来完成某类任务的能在一定的环境中自主发挥作用的、有生命周期的计算实体^[5].

对于构件来讲, 目前采用的比较广泛、全面的定义是: “构件是一个独立发布的功能部分, 可以通过它的接口访问它的服务.”^[6] 目前存在着许多的软件构件模型, 如 EJB, CORBA 等.

在本文中, 把 Agent 设计成构件, 构建的在 Java 环境下运行的 Agent 构件的基本结构如图 1 所示.

依照上述结构, 要实例化一个 Agent, 必须定义好方框内的各个模块. Agent 之间的交流可以通过交流器, 用 Java 的 RMI(远程方法调用) 实现. 各个 Agent 通过对知识的不断更新来提高自己的各种能力.

2.2 由 Agent 构件组成团队的步骤

将上述 Agent 构件组成团队, 步骤如下:

- 1) 假定招聘到的成员 Agent 是有能力工作的, 同时, 每个成员愿意为团队的目标奉献自己的知识和经验.
- 2) 选出一个具有较强的管理能力和人际关系的、愿意为团队作贡献的 Agent 来担当 PMA (Project Manager Agent, 项目管理 Agent).
- 3) 在团队中形成一种使各个成员 Agent 高度依赖的氛围. 团队中的每一个成员都有责任去完成设定的任务, 而且每个成员的工作都是举足轻重的. 它们的每一步工作都会对团队的总体目标产生影响.
- 4) 营造一种轻松的气氛, 使得成员之间能够坦诚地交流, 这对团队的工作非常关键.
- 5) PMA 制定好团队的目标和每个成员的任务.
- 6) PMA 及时发现团队合作中出现的问题并予以解决.
- 7) PMA 鼓励成员的创新并且信任每个 Agent.

3 团队的设计与评价

3.1 Object-Z 简介

Object-Z 是一种形式化说明语言, 常用于需求工程中. 它是 Z 语言的一种面向对象的扩展,

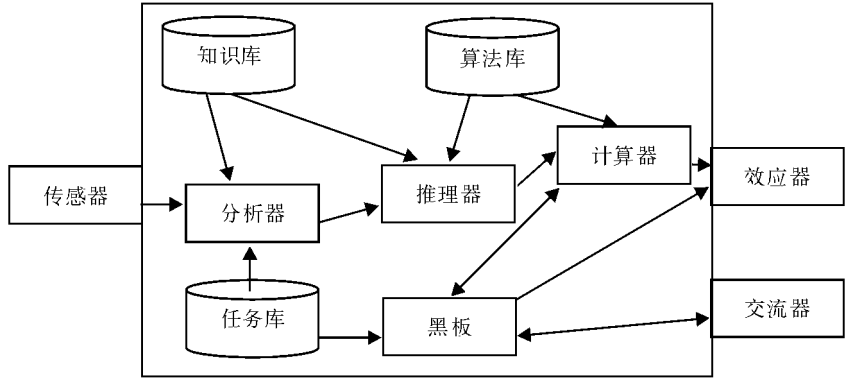


图 1 Agent 的结构

具有形式化方法和面向对象的优点. 它描述的规格说明由类组成, 这些类相互交互共同描述系统的功能. 类由状态及在该状态之上的操作组成. 在 Object-Z 中类用一个命名的方框表示, 它包括 visible list, inherited classes, type definition, constant definitions, state schema, initial state schema, operation schemas, history invariant 等. Visible list 包括了对类中对象的可见操作, 若省略 visible list, 则表示所有的操作都可见.

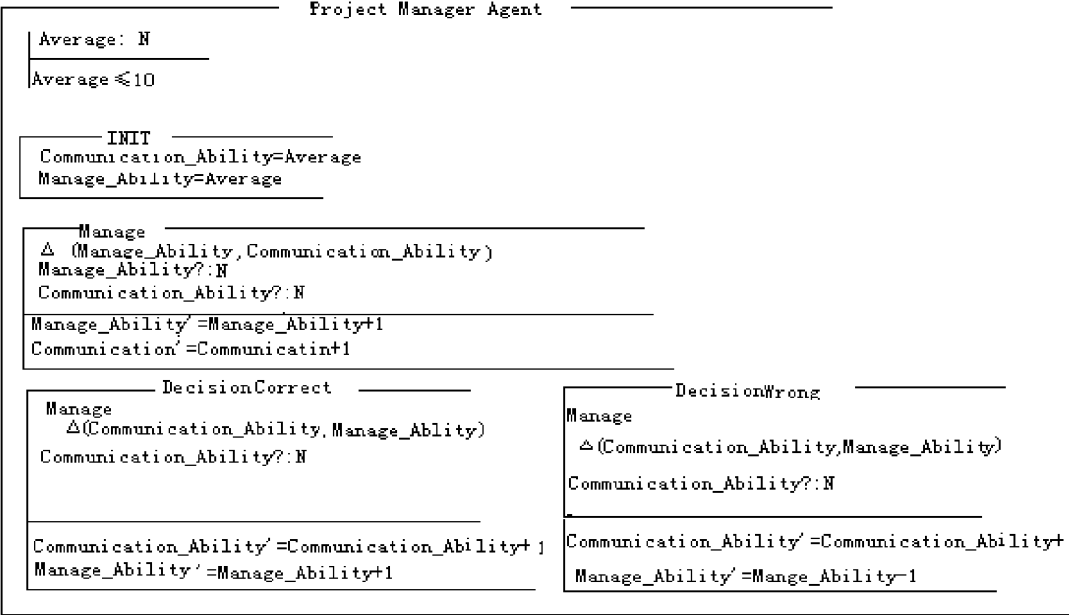
状态模式是匿名的, 它包括了属性声明和相应的状态变量的定义. 常量和状态变量统称为属性. 在常量和状态上的陈述称为类常量^[7]. 某些属性可能依赖其它的属性, 这样的属性称之为: 辅助变量. 初始化模式通常命名为 INIT, 它定义了类的实例的初始状态. 操作模式(Operation schemas), 不同于 Z 的操作模式在于, 有一个 Δ list, 其中包括了个体的状态变量的值变化的对象的集合, 是由上下 2 部分组成的. 上面的部分定义了操作的输入和输出; 下面的部分定义了与操作的初始状态和终止状态有关的谓词.

3.2 用 Object-Z 描述一个分布式团队的规格说明

建立一个小型的团队后, 用 Object-Z 语言描述其规格说明. 对非常小的项目 (1~ 3 人), 项目经理、首席软件架构师等可以合并为一个人^[8]. 在这个系统中需要用到的个体有: PMA 和开发级 Agent(Application Agent), 市场 Agent(Market Agent) 等. 在这个团队中, 用到的数据类型有: 管理能力, 决策能力, 开发能力, 销售能力, 交际能力, 创新能力等. 因此可给出如下的类型声明: [Manage, Decision, Exploit, Sale, Communication, Creation, Sum, Task]

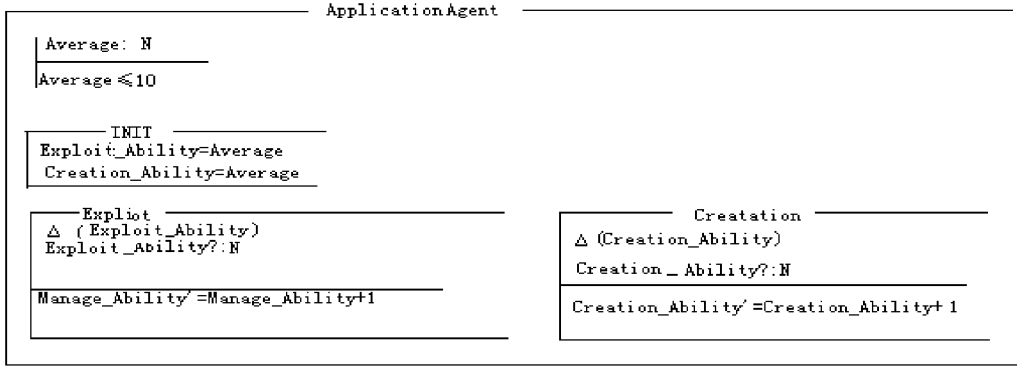
1) 项目管理 Agent(Project Manger Agent)

在 Project Manager Agent 类中, 用到的操作有对其它的 Agent 的管理能力以及对项目的决策能力, 同时应该具有强的交际能力, 使得团队形成良好的工作氛围, 描述如下:



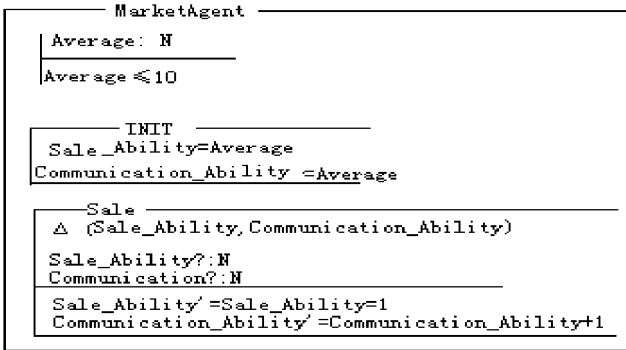
2) 开发 Agent(Application Agent)

在 Application Agent 中, 要求 Application Agent 必需具有很强的开发能力和创新能力:

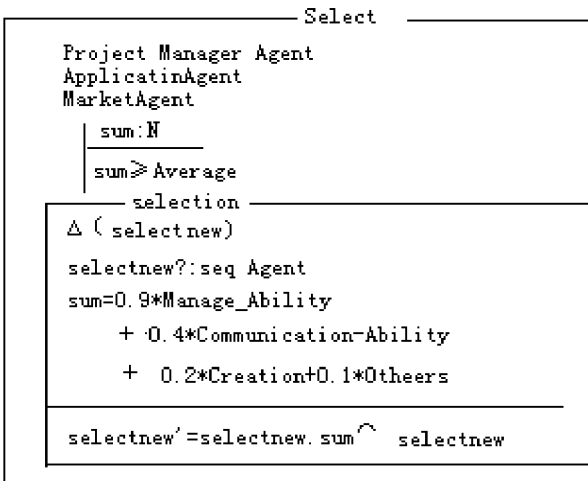


3) 市场 Agent(MA)

在 MA 中, 关注的是 MA 的销售能力和交际能力, 描述如下:



同时还要为团队的成员选举定义一个 Select 方法. 下面以选出一个更合适 Project Manager Agent 来描述一下选举的方法:



因为在选举 Project Manager Agent 时, 最主要的是看它的 Manage_Ability 和它的 Communication_Ability, 所以在求它的能力和(sum)时, 就应该相应地加大这两项能力的系数, 也就是用它的加权能力和.

当然可以让各个 Agent 相互学习, 使得各种能力能够相互补充和完善. 从而可以建成一个更高效率的团队, 来完成任务.

上述团队的构建如图 2 所示.

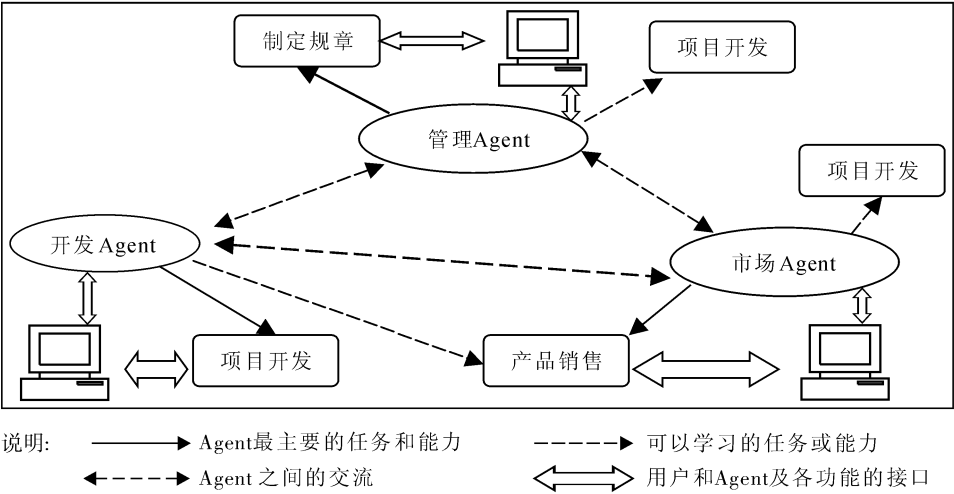


图 2 团队的实现

3.3 团队评价

总结基于多 Agent 的高效率团队的评价方法主要有以下 4 点:

- 1) 各个成员的能力, 特别是 PMA 的能力关系到团队工作的成败.
- 2) 每个 Agent 成员是否愿意为团队共同的任务努力.
- 3) 工作中的冲突如何处理. Agent 成员都要参加处理不同的观点, 以找到办法来更好地完成任务.
- 4) 有没有创新和冒险的能力. 没有创新, 就没有发展, 只有冒险, 不断的尝试错误, 把错误当作是学习的机会, 才能使团队合作得更融洽.

用以上的方法评价我们的设计: 对于本小型团队来讲, 许多 Agent 在实际的工作中可能是一个工作组. 由于 Agent 是个自治的个体, 它就有自己的意愿 (它可以服从管理也可以不服从管理), 这在管理上增加了一定的难度. 解决的方法是给 PMA 最高的权限(如制定大家必须遵守的规则, 迫使其它的 Agent 遵守). 当然, 它们之间的组织关系也包括平等的合作伙伴关系, 而且这些成员以及它们之间的关系也不是一成不变的. 合作期间, 为了让团队有更好的研发氛围和团队的机制不至于太僵化, 在团队建立后的一段时间内, 应该在团队的所有成员中进行投票, 以选出更合适各个 Agent 的职位, 发挥出每个成员的最大功效, 方法是参考各个 Agent 的 sum 用匿名投票的方式进行. 这样组成的团队应该是高效率的.

3.4 基于 Agent 交互团队的价值

所建立的团队, 可以完成大型的分布式的任务, 特别是一些烦杂的、枯燥无味的工作(如: 昼夜不停地谈判; 讨价还价; 搜索市场信息等等). Agent 通过交互, 还能学习以前没有的能力, 以更多的优势增强团队的合作. 对于时间就是效益的今天, 可以节省出时间来做其它的事情, 这的确是很有意义的.

4 结论

面向 Agent 的软件体系结构的研究还处于发展的阶段, 不应认为 Agent 能解决一切问题. 应该看到面向 Agent 的软件工程形成大规模的商业化系统还有很长的路要走, 但它毕竟是软件工

程思想的一种进步. 了解面向 Agent 软件体系结构的团体特征及其存在的极大的优势, 对于扩大面向 Agent 软件体系结构的应用领域有着深远的意义.

参考文献:

- [1] 万建成, 卢雷. 软件体系结构的原理、组成与应用[M]. 北京: 科学出版社, 2002.
- [2] 关旭, 程明, 刘豹. 面向 Agent 软件工程[J]. 计算机工程与应用, 2001, 37(11): 30-32.
- [3] 张友生. 软件体系结构的现状和发展方向[EB/OL]. 软件工程专家网. [2003-03-05]. <http://resource.51cmm.com/NewTech/No040.html>.
- [4] 石纯一, 张伟, 徐晋晖, 等译. 多 Agent 系统引论[M]. 北京: 电子工业出版社, 2003.
- [5] 聂亚杰, 刘大昕. 面向 Agent 的软件工程[J]. 小型微型计算机系统, 2002, 23(4): 417-420.
- [6] Alan W Brown. 大规模基于构件的软件开发[M]. 赵文耘, 张志译. 北京: 机械工业出版社, 2003.
- [7] Gordon Rose. Object Orientation in Z (1992) [EB/OL]. [2005-02-11]. <http://citeseer.ist.psu.edu/rose92Object.html>.
- [8] Daniel J. Paulish 著. 软件项目管理实用指南——以体系结构为中心[M]. 白晓颖, 邵忠焄译. 北京: 机械工业出版社, 2003.

Research on Association Characteristics of Agent-oriented Software Architecture

XIANG Hong-gui, LI Wei-hua

(Faculty of Computer, Guangdong University of Technology, Guangzhou 510090, China)

Abstract: While introducing the current research situation of the software system structure, a group is set up which can reflect users in some ways, especially in the studying and coordinating each other. The characteristics of system structure and Agent-oriented software system structure with Object-Z are described at the same time. The method to set up a high-efficiency group and the rule to appraise it have been put forward.

Key words: agent; object-Z; software engineering; software architecture; team