

上海大学出版社

2005年上海大学博士学位论文 74



基于网格的面向 Agent 软件分析与设计建模 方法及环境研究

- 作者：刘 炜
- 专业：控制理论与控制工程
- 导师：刘宗田



G643/135

上海大学出版社

001280752

2005年上海大学博士学位论文 74



基于网格的面向 Agent 软件分析与设计建模 方法及环境研究

- 作者：刘 炜
- 专业：控制理论与控制工程
- 导师：刘宗田



贵阳学院图书馆



图书在版编目(CIP)数据

2005 年上海大学博士学位论文. 第 2 辑/博士论文编辑部编. —上海: 上海大学出版社, 2009. 6

ISBN 978-7-81118-367-2

I. 2… II. 博… III. 博士—学位论文—汇编—上海市—2005 IV. G643.8

中国版本图书馆 CIP 数据核字(2008)第 180878 号



2005 年上海大学博士学位论文

——第 2 辑

上海大学出版社出版发行

(上海市上大路 99 号 邮政编码 200444)

(<http://www.shangdapress.com> 发行部 66135110)

出版人: 姚铁军

*

南京展望文化发展有限公司排版

上海华业装潢印刷厂印刷 各地新华书店经销

开本 890×1240 1/32 印张 274.25 字数 7641 千

2009 年 6 月第 1 版 2009 年 6 月第 1 次印刷

印数: 1~400

ISBN 978-7-81118-367-2/G·490 定价: 980.00 元(49 册)

Shanghai University Doctoral Dissertation (2005)

Research on Agent-Oriented Software Analysis and Design Modeling Methodology and Environment based on Grid

Candidate: Liu Wei

Major: Control Theory and Control Engineering

Supervisor: Liu Zongtian

Shanghai University Press

• Shanghai •

答辩委员会 上海大学 的评语

本论文经答辩委员会全体委员审查,确认符合上海大学博士学位论文质量要求。

答辩委员会名单:

主任:	尤晋元	教授,上海交通大学	200240
委员:	蒋昌俊	教授,同济大学	200092
	顾 宁	教授,复旦大学	200433
	吴耿锋	教授,上海大学	200072
	李 青	教授,上海大学	200072
导师:	刘宗田	教授,上海大学	200072

学大城土

合科人簡,查审员委村全会员委輯答登文外本

。求要量即文外分半士制学大城土

评阅人名单:

尤晋元	教授,上海交通大学	200240
蒋昌俊	教授,同济大学	200092
胡运发	教授,复旦大学	200433

答辩委员会对论文的评语

网格技术的发展给软件技术提出了新课题,为了适应网格环境的特点,需要相应的新的软件体系结构和软件建模方法。刘炜同学的博士论文“基于网格的面向 Agent 软件分析于设计建模方法及环境研究”在这方面做了系统的研究,主要如下:

1. 以 OGSA 网格体系结构为背景提出了 Agent 抽象机制的分布式软件系统结构以及相应的建模方法——AOMG 方法。该方法用统一的可视化模型语言实现了对网格软件

2. 通过扩展 Object-Z 语言,提出了 Formal AOMG 形式化建模框架,讨论了 AOMG 模型的相关语法及语义特征。

3. 在 AOMG 中采用了基于本体的资源匹配方法,用扩展 UML 类图方法建立了网格资源本体模型,在 UML 和 OWL 之间建立了元模型语义映射关系。

4. 开发了 AOMG 的建模环境,提供了相应的编辑、验证和分析手段。并使用电子政务门户系统为例,验证了所提方法和实现环境的可行性。

答辩委员会认为,论文选题具有重要的学术意义和应用价值,该同学掌握了本学科坚实宽广的基础理论和深入系统的专业知识。

该同学在答辩过程中,叙述清楚回答问题正确。

经过无记名投票,答辩委员会五人一致认为,该论文达

到博士学位论文水平，同意通过答辩，建议授予刘炜同学工学博士学位。

答辩委员会成员名单

答辩委员会委员：蒋昌俊

2005年5月18日

摘 要

网格是一种由硬件和软件构成的信息技术基础设施,能提供可靠的、可协调的、可扩展的和廉价的高端计算能力的访问,并支持动态的、分布式的虚拟组织之间不同资源的共享和协作。基于网格的问题求解称作网格计算,网格计算是近年来兴起的一个研究热点。网格计算环境具有异构性、可扩展性、动态自适应性等特点,这些特性使得运行在网格平台上的软件系统不同于在集中和封闭环境下发展起来的传统软件形态,将呈现出以自主性、协同性、开放性以及互操作性为基本特征的全新形态。我们将这种新的软件形态定义为网格软件,或称网格分布式系统。

软件建模技术的发展是通过不断提出更具表达能力、更符合人类思维模式的抽象机制来为复杂系统建立模型。过程抽象、抽象数据类型、以及目前比较常用的“对象”和“组件”都属于这样的抽象机制。目前在软件领域占主导地位的软件方法是面向对象方法,但是面向对象软件技术体系其本质上是一种静态和封闭的框架体系,难以适应网格环境开放、动态和多变的特点,因此,网格软件这种新型的软件形态对传统的软件理论、方法、技术和平台提出了一系列挑战。

从分布式人工智能观点考虑,网格计算和多 Agent 系统具有很强的相似性。最明显表现在多 Agent 系统中的实体和网格实体都具有自治性;Agent 还具有其他一些和网格实体相关

的特性,如社会能力、自适应性和自发性等;同时,Agent 和网格实体一样具有异构性,不同的 Agent 可以由不同的程序设计语言、体系结构和技术实现。Agent 的这些特点使之成为研究网格实体的一种重要的抽象机制。

当前主要的面向 Agent 建模技术基本上是针对传统的封闭式软件体系结构而提出,对于网格软件的建模缺乏有效的概念抽象机制和直观的模式描述。基于对现有网格技术和主要面向 Agent 软件工程方法的研究,我们选择 OGSA 开放式网格体系结构作为网格软件的运行平台(或称应用背景),依照面向 Agent 软件工程的要求,提出以 Agent 作为基本单元的分析策略,形成了 AOMG(Agent-Oriented Modeling based on Grid)软件建模方法,能够有效地建立基于网格的新型分布式系统模型。该方法的研究内容主要包括:

(1) 提出了以 OGSA 网格体系结构为应用背景,基于 Agent 抽象机制的分布式软件系统结构。

(2) 在传统 Agent 定义的基础上,结合网格软件形态的一些特征和 OGSA 结构的以服务为核心的基本思想,对 Agent 概念重新定义,并引入虚拟组织、开放式组织、网格 Agent (GridAgent)和 Agent 服务等新的抽象概念,为网格软件的分析 and 设计建模提供概念抽象机制。

(3) 通过改进现有的 i^* 建模框架以及对 UML 的扩展,结合组织建模思想,AOMG 提供了一套适合描述网格分布式应用的可视化建模框架,通过统一的可视化模型语言实现了对网格软件的直观描述。在建立模型语言的同时,AOMG 建模框架提出了基于元模型层次的模型变换规则,很好地解决了需求分析

过程中的 Agent 抽取问题,缩小了设计过程中从 Agent 抽象层次到对象层次之间的鸿沟。

(4) 任何一种系统化的建模方法,都必须建立在对建模概念和基本模型的语法和语义的精确定义的基础上。为此,在 AOMG 建模概念基础上,通过扩展 Object-Z,我们提出了 AOMG 形式化建模框架,Formal AOMG。该框架为 AOMG 建模方法中各种概念和元模型提供了精确无二义性的定义,同时为 AOMG 可视化建模框架中的模型变换规则提供了基于精确语义描述的形式化基础。Formal AOMG 既可以作为独立的针对网格软件的形式化建模语言,也可以作为 AOMG 建模方法在后续设计、实现和模型验证过程中的一种补充。

(5) 对于网格软件来说,为了实现服务的共享,如何在网格环境中选择资源以及如何为其他网格应用提供资源是需要解决的重要问题,因此需要提供一种有效的资源匹配机制。在 AOMG 建模方法中,我们采用了基于本体的资源匹配方法。通过扩展 UML 类图来建立网格应用的资源本体模型,而为了准确快速地构造基于 OWL(Web Ontology Language)的可用本体,我们在 UML 和 OWL 之间建立一个元模型语义映射关系表。

(6) 为了验证 AOMG 方法体系中各种建模思想以及模型语言的有效性,我们开发了一个基于 AOMG 方法体系的建模环境——AOMG Toolkits。AOMG Toolkits 建模环境提供了可视化的模型编辑器以及基于元模型层次的模型变换工具,实现了基于 OWL 的模型存储,同时通过元模型的刻面约束机制实现了 AOMG 元模型语法的正确性检查。

(7) 通过一个电子政务网格门户系统完整地演示了网格软件系统的需求获取、分析模型和设计模型的建立(包括基于形式化框架的模型建立)、Agent 的抽取、分析模型到设计模型的映射、资源本体的建立等内容。

关键词 网格计算, Agent, 网格 Agent, 网格服务, AOMG 建模方法, Object-Z 语言, 形式化建模框架, 本体, OWL, 模型映射规则, 模型语法检查

Abstract

“Grid” is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities. Grid coordinates resource sharing and problem solving in dynamic, multi-institutional virtual organizations. Grid-based Problem Solving is called grid computing, which has emerged as a research hot spot field currently. Distributed environment based grid computing has characteristics such as heterogeneity, extensibility, openness and adaptability. Therefore, software systems based on grid infrastructure are different from software systems based on conventional distributed environment. It is to appear autonomous, cooperately, open and interoperable. We called this software form *Grid Software*, or *Grid Distributed System*.

Progress in software modeling technologies over the past two decades has been made through the development of increasingly powerful and natural high-level abstractions with which to model and develop complex systems. Procedural abstraction, abstract data types, and, most recently, objects and components are all examples of such abstractions. Object has become the chief paradigm in software engineering field, whereas, object-oriented modeling method is a static and

close technology framework in essence. From the perspective of *Grid Software*, object-oriented methods are difficult to adapt well to the features of grid software. The development of grid computing has challenged conventional software architectures, methods, technologies and platforms.

If grid computing is considered from a Distributed Artificial Intelligence (DAI) perspective, it is similar to a Multi-Agent System (MAS), since entities in a MAS and the grid have autonomous behavior. Autonomy is one of the key abstraction features of agents. Other features of agents relevant to grid entities include social ability, as well as adaptability and pro-activeness. Different in addition, software agents are heterogeneous, which could be implemented in different programming languages and technologies. These characteristics of agents justify the selection of agents as the main abstraction for modeling grid software.

Most of today's agent-oriented modeling methods emphasize modeling conventional close software architecture, and while model a grid software, they lack of effective concept abstraction and declarative models. Based on the research of grid architecture and main agent-oriented software modeling methods, this thesis puts forward a new modeling method for agent-oriented analysis and design based on OGSA architecture, AOMG (Agent-Oriented Modeling based on Grid). The main research contents of AOMG method include:

1. Proposes a multi-agent architecture of the distributed software that based on OGSA grid infrastructure.
2. Based on traditional definition of agent, incorporating the characteristics of Grid Software and the idea of grid service from OGSA, AOMG redefines the concept of agent, as well introduces some new abstraction concepts (such as *Virtual Organization*, *GridAgent* and *GridService*) related to grid architecture. These new concepts provide an abstraction mechanism for modeling grid software.
3. By improving i^* framework and extending UML, and taking viewpoints from Organization, AOMG presents a visual modeling framework for grid software. At the same time, AOMG provides model mapping rules based on meta-model level, which solve the problem of agent's elicitation during requirement analysis stage, and narrow the gap between analysis models at agent-level and design models at object-level.
4. At the basis of AOMG modeling theory, by extending Object-Z language, this thesis proposes a formal modeling framework, Formal AOMG. This formal framework provides meanings for common concepts and terms of AOMG precisely and unambiguously, and does so in a readable and understandable manner. As well, it provides a foundation for subsequent development of new and increasingly more refined concepts, as well provides formal foundation for models' transformation and development of visual modeling environment.

5. In order to enable resource sharing and coordinated problem solving in dynamic multi-institutional virtual organizations, it's necessary to solve the resource matching problem. In AOMG method, we propose an ontology-based resource matching method, and present a resource ontology model by extending UML class model. A meta-model mapping rule table between UML and OWL is proposed to facilitate the construction of OWL-based resource ontology.

6. In order to test the validity of the above methods, we develop a modeling environment based on AOMG method, AOMG Toolkits. This modeling environment supports developer to model system visually, analyze and check models. AOMG Toolkits store models in OWL format, and check model syntax by introducing meta-model's facet constraint.

7. Through an e-government grid portal application example, we demonstrate the requirement acquirement of grid software, the creation of analysis models and design models (as well formal models), elicitation of agents, the mapping from anlysis models to design models and the creation of resource ontology model.

Key words Grid Computing, Grid Software, Agent, Grid Agent, Grid Service, AOMG, Object-Z, Formal AOMG, Ontology, OWL, Meta-Model Mapping, Model Syntax Checking

目 录

第一章 绪论	1
1.1 网络计算与网络软件	1
1.1.1 网络概念及其特点	1
1.1.2 网络体系结构	2
1.1.3 基于网络体系结构的软件形态	4
1.2 基于网络的软件建模与 Agent 抽象机制	6
1.3 主要的面向 Agent 软件工程方法和建模工具	8
1.3.1 面向 Agent 软件工程方法	8
1.3.2 面向 Agent 软件开发环境	13
1.4 研究意义和内容	15
1.5 论文组织结构	18
第二章 AOMG: 一种基于网络计算的面向 Agent 建模方法	20
2.1 与 AOMG 相关的研究与技术	20
2.1.1 OGSA 网络体系结构	20
2.1.2 i* 策略建模框架	24
2.1.3 UML 建模语言	26
2.2 基于 OGSA 的面向 Agent 可视化建模框架	27
2.2.1 基于 OGSA 的软件体系结构	27