

# 协同联盟中组织模型的研究<sup>\*</sup>

路 鹏 张小蕾 陶先平 吕 建

(计算机软件新技术国家重点实验室 南京大学计算机软件研究所 南京210093)

## The Study on the Organizational Model of Coordination Association

LU Peng ZHANG Xiao-Lei TAO Xian-Zhi LU Jian

(State Key Laboratory for Novel Software Technology, Institute of Computer Software, Nanjing University, Nanjing 210093)

**Abstract** The architecture of a coordination system can naturally be viewed as an organized society of individuals (i. e., as a computational organization). For this reason, we believe organizational model should play a central role in the coordination systems including coordination association. To this end, the concepts of role models, architecture styles, and coordination laws are increasingly being used to specify and design coordination systems. However, this is not the full picture. In this paper we develop three organizational concepts - organizational role model, organizational structure, and organizational rules - that we believe are necessary for the complete specification of computational organizations of coordination association.

**Keywords** Coordination association, Organizational model, Role model

## 1 引言

随着 Internet 的快速发展和普及,应用网络化已成为计算机应用的主流。在应用网络化的基础之上,“软件作为服务”的趋势深刻影响软件技术的未来,在该理念的驱动下,Web Service 概念及相关技术应运而生并得到了迅速发展,成为当今网络技术的热点。

从技术角度,开放的网络化应用和“软件作为服务”这一新模式的推行以及 Web Service 技术的发展必将导致 Internet 环境下软件系统的主要形态、运行方式、生产方式和使用方式发生巨大的变化。未来的网络软件系统将体现为具有内在协作关系的若干 Web Service,应用的满足将通过 Web Service 的协同来完成。然而,目前 Web Service 及其协同技术的研究主要还是局限在软件服务标准的制定、推广以及在接口和数据表示层次上的互操作技术,但是软件服务协同的研究远不仅限于此,服务的功能互通、协作联盟以及多模式协同的研究更具挑战性。

一般来讲,软件服务是一个相对独立的计算实体,通常由处理逻辑和协同逻辑两部分组成,前者描述软件服务的功能,后者描述与其他软件服务之间的协同。然而,软件服务的协同模式是多样的,如以 CORBA 为代表的 RPC 模式、以 JavaSpace 为代表的共享空间模式,因此,在考虑到功能互通因涉及语义而产生的困难基础上,我们认为:多模式的软件协同及协同联盟将成为下一代开放化网络技术的核心技术之一。而与此同时,90年代中期出现的软件 agent 技术因其移动性和协同性可为软件服务间的多元化协同提供直接的支撑<sup>[1,2]</sup>。基于 agent 的软件协同技术在支撑适应网络环境的多元化模式和协同模式的统一实现框架方面具有较大的发展潜力。

为实现多模式软件服务间的高层次互操作,我们利用软

件 agent 的封装能力,将软件服务的协同逻辑封装为协同 agent。这样,每个软件服务将由一个功能部件(处理逻辑封装体)和一个协同 agent 构成,而网络应用软件将由一组具有上述结构的软件服务组成,网络软件运行将体现为协同 agent 的迁移及其间的交互和软件服务功能部件的执行。基于上述分析,我们给出了一个基于 agent 的软件服务协同抽象模型(图1)。

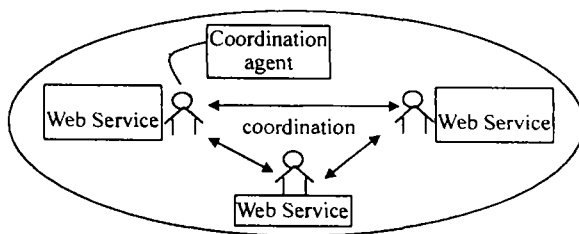


图1 软件服务协同模型

此模型中,一个协调一致、高效管理的协作联盟是保障这些软件服务有机协同的关键。在协作联盟的研究工作中,组织概念及其相关技术引起了人们的普遍关注,越来越多的学者认为:组织模型是一种良好的协作联盟的体现,它可以有效承担起组织内软件服务协同的管理工作。

文[3]中认为协同包含三个要素:实体、载体和法则。其中协同的实体是进行协同交互的 Agents、进程、对象、构件等元素;协同的载体是协同机制的一种抽象,指的是通过什么方式进行协同;协同的法则负责管理实体间的协同交互和载体的行为。将这三个要素对应到组织模型中来:协同的实体可以担任组织的成员,行使组织特定职位上的角色任务;协同机制得到了更进一步的抽象,组织成员是通过特定的组织结构方式来进行协同,所有交互都按照结构关系来进行,可以认为协同

<sup>\*</sup> 本文受国家杰出青年基金项目(61525204)、江苏省高科技项目(BG2001012)和十五863计划高科技项目(2001AA113110)资助。路 鹏 硕士生,研究领域:对象技术,移动 agent 技术。张小蕾 硕士生,研究领域:对象技术,移动 agent 技术。陶先平 副教授,博士,研究领域:对象技术,移动 agent 技术。吕 建 博士,教授,博导,研究领域:形式化方法,对象技术,分布计算技术和构件技术。

的载体抽象到了组织结构一层;协同的法则也演化为组织规则,对组织成员间的任何交互进行约束和管理。

本文围绕上述软件服务结构,针对协同和组织模型,首先讨论了该方向的研究现状,包括面向 Agent 的分析和设计方法学 Gaia<sup>[4]</sup>中的角色模型;基于目标(Goal-Based)的组织形态分类和描述<sup>[5,6]</sup>;依据非功能性需求(Non-Functional Requirements)<sup>[7]</sup>对组织结构的评价和选取;Internet Agents 协同模型及协同法则的比较<sup>[3]</sup>,组织规则的形式化描述<sup>[8]</sup>等,然后进行了上述模型的特点分析,最后结合我们正在开展的基于 agent 软件协同研究工作,给出了一个统一的组织模型,其中包括组织模型的角色模型描述、结构的描述方法、结构形态多角度的分类以及规则池的概念及描述。

## 2 研究现状

国际上协同领域及组织领域的研究近年来有很多论述,我们依据协同联盟的要求选择了其中有代表性的加以研究。

### 2.1 角色模型

在文[4]中,提出了面向 Agent 的分析和设计方法学 Gaia,其中对协同实体的角色模型作了很好的总结与论述,一个典型的角色应涉及以下内容:名字,自然语言描述,功能,权力和职责。下面是 Gaia 角色模型的模板:

Role Schema:	name of role
Description	short English description of the role
Protocols and Activities	protocols and activities in which the role plays a part
Permissions	"rights" associated with the role
Responsibilities	
Liveness	liveness responsibilities
Safety	safety responsibilities

我们可以用一个角色实例来深入理解,下面的“添咖啡器”这一角色保证向咖啡壶中不断地添加咖啡,当咖啡充好后通知工人。

Role Schema:	CoffeeFiller
Description	This role involves ensuring that the coffee pot is kept filled, and informing the workers when fresh coffee been brewed.
Protocols and Activities	Fill, InformWorkers, CheckStock, AwaitEmpty reads supplied coffeeMaker // name of coffee maker
Permissions	coffeeStatus // full or empty changes coffeeStock // stock level of coffee
Responsibilities	
Liveness	CoffeeFiller = (Fill, InformWorkers, CheckStock, AwaitEmpty) $\omega$
Safety	coffeeStock > 0

我们可以看到这种角色模型对实体本身的内容及对外的接口均有完备的描述,但从组织的研究来看,有下面几点不足:抽象程度仍停留在角色一级,无组织信息的描述,抽象级别不够;角色自身行为和协同行为、协同规则合为一体,灵活性不够。

### 2.2 组织结构

结构是组织最重要的特性之一,是无序到有序的关键,将个体间原本自由的协同和交互按照结构组织起来进行,才能

有效地达到整体的功能目标。

文[5,6]中对基于目标(Goal-Based)的组织结构形态做了深入的研究,提出了 Structure-in-5, Joint Venture, Bidding, Coopation ...等结构形态,并给出了结构的描述方法,下面我们看一下最典型的 Structure-in-5结构(图2),它由许多组织中都可以见到的一些策略和逻辑构件组成。最下层是 Operational Core,它负责执行最基本的任务和操作,如输入、处理、输出等;最上层是 Apex,它代表了策略决策人员;中间是一些控制、管理构件和逻辑部分,由 Coordination, Middle Agency 和 Support 组成。

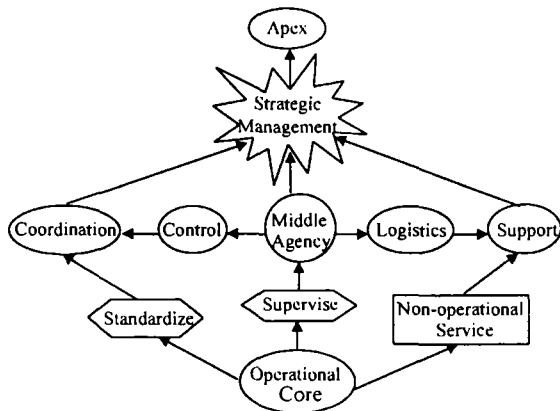


图2 一种组织结构形态 Structure-In-5<sup>[5]</sup>

图中圆圈内的 Apex, Coordination, Middle Agency, Support, Operational Core 为五个协同实体,描述如下:

```

TELL CLASS StructureIn5MetaClass
  IN Class WITH /* Class is here used as a MetaMetaClass */
  attribute
    name: String
  part, exclusivePart, dependentPart
  ApexMetaClass: Class
  CoordinationMetaClass: Class
  MiddleAgencyMetaClass: Class
  SupportMetaClass: Class
  OperationalCoreMetaClass: Class
END StructureIn5MetaClass
  
```

它们之间的协同关系有 taskDepend, goalDepend, softgoalDepend,再根据依赖与被依赖的关系,整个结构的协同关系描述如下:

```

TELL CLASS CoordinationMetaClass
  IN Class WITH /* Class is here used as a MetaMetaClass */
  attribute
    name: String
  taskDepended
    s: StandardizeTask
    WITH dependee
      OperationalCoreMetaClass: Class
  END
  goalDepended
    c: ControlGoal
    WITH dependee
      MiddleAgencyMetaClass: Class
  END
  softgoalDepender
    s: StrategicManagementSoftGoal
    WITH dependee
      ApexMetaClass: Class
  END
END CoordinationMetaClass
  
```

同时,非功能性需求(Non-Functional Requirements)的研究给出了分析设计阶段中组织的结构如何选取的方法<sup>[7]</sup>,如图3所示:软件质量有如下性能的需求 Availability, Security, Adaptability 等,每种需求又可继续细分;而各种不同的组织结构对这些性能的贡献不同,有的结构加强了某性能,同时又削弱了另外一个性能。通过对结构的综合评价,可以得出不

同软件质量侧重下的最佳组织结构。

	Structure-In-5	Pyramid	Bidding	...
Predictability	+	++	--	...
Security	+	++	--	...
Adaptability		+	++	...
Cooperativity	+	++	-	...
Competitivity	-	-	++	...
Availability	+	+	-	...
Failability-Tolerance		--	--	...
Modularity	++	-	++	...
Aggregability	++			...

图3 组织结构的评价及选取<sup>[5]</sup>

	Coordinables	Coordination Media	Coordination Language	Coordination Laws
JavaSpaces	Java Agents	OO referencing	OO (Java)Linda primitives	Fixed (object-matching)
Lime	Abstract mobile components	Implicit access (to agent-own-tuple space)	Linda	Fixed (pattern-matching)
MARS	Java mobile agents	Implicit access (to local tuplespace)	OO (Java)Linda primitives	Programmable (Java reactions)
PageSpace	Agent + Services	OO referencing	OO (Java)Linda primitives	Fixed (object-matching)
T Spaces	Agent, PDA	OO referencing	Linda + user-defined primitives	Programmable ( primitivesoverriding)
TuCSoN	Heterogeneous (possibly mobile)Internet agents	Implicit access (to local tuplespace) + URL referencing	Logic-oriented Linda	Programmable (FOL reactions)

图4 Internet Agents 协同的比较<sup>[3]</sup>

在文[8]中,更是对组织规则的形式化描述作了初步的探索,文章中定义了如下时序算子:

Formula	Interpretation
$\bigcirc\varphi$	$\varphi$ is true next
$\diamond\varphi$	$\varphi$ is eventually true
$\square\varphi$	$\varphi$ is always true
$\varphi U \psi$	$\varphi$ is true until $\psi$ is true
$\varphi W \psi$	$\varphi$ is true unless $\psi$ is true
$\varphi B \psi$	$\varphi$ is true before $\psi$ is true

下面是用形式化描述的一些组织规则,其中谓词 plays (i,r)表示实体 i 扮演角色 r:

- 任何一个实体都不能同时扮演角色 r1 和 r2  
 $\forall i \rightarrow [Plays(i, r1) \wedge Plays(i, r2)]$
- 直到有了实体扮演角色 r2才允许角色 r1被扮演  
 $\exists i \cdot Plays(i, r1) U \exists j \cdot Plays(j, r2)$
- 一旦实体扮演了角色 r1,它将永远扮演下去  
 $\forall i \cdot Plays(i, r1) \Rightarrow \square Plays(i, r1)$

从前面的介绍可以看出,协同模型的法则并没有过多涉及组织的概念;而形式化描述是一个很好的尝试,但没有和角色模型及组织结构统一起来。

### 3 协同联盟中的组织模型

因为软件环境 Internet 复杂多变,协同联盟系统中的组织应做到角色、结构、规则相对独立,即同一种角色可以存在于不同的结构中,组织可以动态地改变协同的规则,同一个组织可以动态改变成员角色、组织结构等等,实现协同联盟的开放、动态、灵活等特点。

上一章分析了一些研究现状,针对我们自己的系统,协同

我们可以看到,上述研究给出了组织结构分类的一种有益的角度(Goal-Based),及结构的描述方法,同时还有如何在不同的结构中进行选取以达到组织最佳的性能,但是,结构分类的角度较为单一,结构的描述方法有局限也是需要解决的问题。

### 2.3 组织规则

组织规则是由协同法则逐渐发展来的,图4列举了几个目前具有代表性的 Internet 协同模型<sup>[3]</sup>,里面提到的可编程的协同法则对组织规则有很好的借鉴作用。

联盟中对角色模型、组织结构、组织规则均作了一些改进,下面依次论述。

### 3.1 角色模型

这里的角色模型是指组织这一抽象级别下的成员模型,即对组织内的职位作一定义,是一种抽象,而实体(Agents,构件等)如满足模型要求,将来可以担任此角色,成为组织的成员。

在组织的层次上,需要对角色模型所应包含的内容做一界定,以满足协同联盟中组织灵活性的要求。我们对 Gaia 的角色模型修改后,得到协同联盟的组织角色模型。

Role Schema:	name of role
Organization Ref.	belong to which organization
Description	short English description of the role
Activities	individual activities in which the role plays a part
Protocols	activities requiring interaction with others

可以看到,角色模型中增加了组织索引,而权力和职责部分被去掉了:

①组织索引是不可少的,通过它才能确定角色和其它角色间的依赖关系及相互协同时所遵循的规则;而且通过修改组织信息,该角色可重用到不同组织中。在我们的角色模型中,这一部分是一个索引集合,即一个角色可以同时属于多个组织。

②Activities 部分是角色固有的,予以保留。

③Protocols 是该角色与其他角色需要进行交互协同才能完成的行为,要受到组织规则的约束,因此 Protocols 与角色间的依赖关系(见第二节)相对应,描述为下面的二元组:(Protocol, Dependency Ref.),其中 Dependency Ref. 指向协

同关系,见下一节。

④权力和职责部分可以说是组织所赋与的,一个角色能从组织得到什么、或应为组织做出什么是组织的特性,而不是角色本身固有的,因此我们把它们放到组织规则中去,既保持了角色的相对独立,又满足了动态修改权力和职责的灵活性,及有效的监督和管理。

实体通过扮演角色加入到组织中来进行协同,而通过角色模型可以实现对实体功能的裁减和组合:因为 Internet 上实体的行为是不可预测的,除去组织内需要的功能,一个实体可能还有其他多样的行为,一旦通过角色行为来约束和裁减实体的功能,实体在组织内的行为就变得可控了;还有一种情况很普遍,组织角色的行为不能由单一的一个实体来完成,或找不到可以完全胜任角色的实体,而多个实体的不同功能组合起来可以满足角色的需求,这样通过一种封装机制,就可以把不同的实体组合到一起扮演一个角色,进行有效的协同。

### 3.2 组织结构

本节先讨论组织及结构模型的描述,然后从多种角度对结构形态进行一种分类。

(1) 每个角色都隶属于一个组织,这就是角色模型中的组织索引,组织的定义则应包含如下内容。

Organization Schema:	<i>name of organization</i>
Description	<i>short English description of the organization</i>
Organization Structure	<i>organized by which type of structure</i>
Rules Pool Ref.	<i>ruled by which set of rules</i>

“组织描述”用自然语言对组织的职能简单介绍;“组织结构”描述了组织成员的组成和之间关系的静态图画;“规则池”对组织如何正常地动态运作进行管理和约束。

这一节具体讨论组织结构,规则池的讨论放在下一节。

组织结构的定义如下:

Organization Structure	
Meta Structure	<i>inherited from which structure style</i>
Composition Def.	<i>include which roles</i>
Dependency Def.	<i>relationships in all of the roles</i>

元结构是我们预定义的各类结构形态,如上一章提到的 Structure-In-5等,我们从多种角度对结构形态进行了分类,在本节的后半部分讨论。

结构组成定义简单列举了整个组织中涉及的所有角色,类似上一章中 Structure-In-5的协同实体的类描述。

Composition Def.
Role 1
Role 2
... ..
Role n

结构依赖关系定义不同于上一章的协同关系的类描述,不再是简单的 taskDepend, goalDepend, softgoalDepend 等几种协同关系,而是和协同行为相对应,定义为下面的四元组:

Dependency Def. = (Protocol Ref., Start Role Ref., {End Role Ref.}, {Rule Ref.})

即

协同关系定义=(协同行为 ref., 发起角色 ref., {接收角色 ref.}, {规则 ref.})

①其中的 Protocol Ref. 为协同行为的指针,作为协同关系的标识,该协同行为在角色模型中定义;

②每一个协同行为都涉及角色间方法的调用或数据的流动,我们可以根据这一点来定义发起角色和接收角色,考虑到组播,接收方定义为角色集合,这样对协同行为的实施可以检验其角色的正确性;

③对协同行为的约束由规则池中的规则来实现,我们需要执行的多条规则的索引放在这里,实现行为和规则的分离,当需要修改规则、或增加删除规则时,只需要添加或删除索引即可。

(2) 组织结构形态的分类是多种多样的,上一章介绍的是基于目标的角度,并使用非功能性需求的理论对结构形态进行评价和选取。下面我们引入了多种角度的分类,供以后选取和研究:

i)按协同方式分为合作和竞争:如 Joint Venture 结构, Bidding 结构等;

ii)按结构尺度从小到大分为:点到点的传统 C/S 结构,多 Agents 系统,超大规模组织等;

iii)按结构形态的相似性:左右对称,中心对称,自相似等;

iv)按微观行为:Broker, Matchmaker, Monitor, Mediator 等;

v)环境相关的协同(如环境的安全性限制)和应用相关的协同(应用逻辑),其结构的外延和嵌套也需要研究。

### 3.3 组织规则

组织规则在协同联盟系统中的作用是巨大的,它对组织如何正常地动态协同进行管理和约束。我们引入了规则池(Rules Pool)的概念,任何一个协同行为都要通过规则池,由其中的规则进行约束,判断是否允许执行。这样,我们实现了组织层次上的集中管理,当然,在可控制的范围内实施多层次集中管理,形成规则池的树结构。

以往,是以角色为中心,该做什么、不该做什么由角色来承担;在我们系统中,是以组织为中心,一切协同规则都是为了达成组织的整体目标,担任某角色的成员也许会意图做出不可预测的行为,但在我们系统中可以保证整体行为的有效。这样既可适应 Internet 环境下开放、动态的情况,又可以使系统的可用性、安全性得到实施。

还要说明的是,组织规则和组织结构是密切相关的,可以说是动态和静态的关系,二者是相辅相成又相对独立的。

规则池定义如下:

Rules Pool:	<i>name of rules pool</i>
Organization Ref.	<i>belong to which organization</i>
Description	<i>short English description of the rules</i>
Rules	<i>corresponding to protocol activities</i>

组织索引和前面一样,与角色和结构建立联系,但规则池的复用还有待研究。

规则池是由规则组成,针对协同行为规则定义如下 Rule = (Pre, Post),凡是协同行为满足规则前置条件的就允许执行,满足后置条件的才能继续执行角色的下一个行为。整个组织的运行启动也是由前置条件所引发的,即任何担任角色的实体,如果它的第一个行为满足我们的前置条件,就由组织负责激活。

上一节协同关系中 Rule Ref. 所指引的就是对协同行为

进行条件检查的规则。另外, Gaia 模型中的角色权力和职责也融入到这一部分中, 这样把传统中和角色紧密相关的一些规则剥离开, 提升到了组织规则中来。当我们要全局修改一些协同规则时, 可以方便地在规则池中进行添加、删除或修改, 不需要对角色或结构进行全局改动, 提高了系统的灵活性。

进一步来说, 规则虽然约束的是角色间的协同行为, 但我们规则池中的规则不涉及具体的角色或行为。这样同一条规则可以被不同的协同行为所指引, 对协同进行约束, 实现规则的复用。每一个协同行为可以指向多个规则, 如果一个角色 Protocol Activity 的协同规则需要进行动态的变动, 则只需向规则池中添加没有的规则, 并改变与该 Protocol Activity 相关的 Rule Ref. 即可。

规则的描述可使用类似 Tuple Space 和 Linda 描述的方法<sup>[3]</sup>。

**结束语** 本文回顾了目前国际上协同和组织研究领域的一些工作进展, 并针对我们的协同联盟系统作了一些改进工作, 提出了协同系统中组织模型的概念, 包括组织角色模型, 组织结构, 组织规则三部分主要内容, 并给出了具体描述方法。

协同联盟组织模型的研究还有很多内容, 怎样解决组织的角色和实体 (Agents, 构件等) 的匹配问题, 如何使角色更好地适应组织规则, 如何重用组织规则, 同时也需要加强对协同技术的研究。

## 参考文献

- 1 陶先平, 吕建. 一种新的移动 Agent 结构化迁移机制的设计和实现. 软件学报, 2000, 11(7): 918~923
  - 2 吕建, 张鸣, 廖宇, 陶先平. 基于移动 Agent 技术的构件软件框架研究. 软件学报, 2000, 11(8): 1018~1023
  - 3 Zambonelli F. Coordination Models and Technologies for Internet Agents. In EASSS: Coordination of Internet Agents, August 2000
  - 4 Wooldridge M, Jennings N R, Kinny D. The Gaia Methodology for Agent-Oriented Analysis and Design. Autonomous Agents and Multi-Agent Systems, 2000, 3(3): 285~312
  - 5 Kolp M, Castro J, Mylopoulos J. A Social Organization Perspective on Software Architectures. In: Proc. of the First Intl. Workshop From Software Requirements to Architectures (STRAW'01), Toronto, May 2001
  - 6 Yu E. Modelling Strategic Relationships for Process Reengineering: [Ph. D. thesis]. Department of Computer Science, University of Toronto, Canada, 1995
  - 7 Chung L K, Nixon B A, Yu E, Mylopoulos J. Non-Functional Requirements in Software Engineering, Kluwer Publishing, 2000
  - 8 Zambonelli F, Jennings N R, Wooldridge M. Organisational Rules as an Abstraction for the Analysis and Design of Multi-Agent Systems. IJSEKE, 2001, 11(3): 303~328
- 
- (上接第107页)
- 3 Cohen W W. Fast effective rule induction. In: Machine Learning: the 12<sup>th</sup> Intl. Conf. Lake Tahoe. CA. 1995. Morgan Kaufmann, 1995
  - 4 Lee W, Stolfo S J. Data mining approaches for intrusion detection. In: Proc. of the 7<sup>th</sup> USENIX Security Symposium. San Antonio. TX. Jan. 1998
  - 5 Lee W, Stolfo S J, Chan P K. Learning patterns from unix processes execution traces for intrusion detection. In AAAI Workshop on AI Approaches to Fraud Detection and Risk Management. AAAI Press, 1997. 50~56
  - 6 Quinlan J R. Learning logical definitions from relations. Machine Learning, 1990, 5: 239~266
  - 7 Chosh A, Schwartzbard A. A study in using neural networks for anomaly and misuse detection. In: Proc. of the 8<sup>th</sup> USENIX Security Symposium, 1999
  - 8 Agrawal R, Srikant R. Fast algorithms for mining association rules. In: Proc. of the 20<sup>th</sup> VLDB Conf. Santiago. Chile. 1994
  - 9 Lee W, Stolfo S J. A data mining framework for building intrusion detection models. In: Proc. of the 1999 IEEE Symposium on Security and Privacy. May 1999
  - 10 Chan P K, Stolfo S J. Toward parallel and distributed learning by meta-learning. In AAAI Workshop in Knowledge Discovery In Databases. 1993. 227~240
  - 11 Mannila H, Toivonen H, Verkamo A I. Discovering frequent episodes in swquences. In: Proc. of the 1<sup>st</sup> Intl. Conf. on Knowledge Discovery in Databases and Data Mining. Montreal, Canada. Aug. 1995
  - 12 Ghosh A K, Schwartzbard A, Schatz M. Learning program behavior profiles for intrusion detection. In: Proc. of the 1st USENIX Workshop on Intrusion Detection and Network Monitoring. USENIX Association, April. 1999
  - 13 Ghosh A K, Schwartzbard A, Schatz M. Using program behavior profiles for intrusion detection. In: Proc. of the SANS Intrusion Detection Workshop, Feb. 1999
  - 14 Ghosh A K, Wanken J, Charron F. Detecting anomalous and unknown intrusions against programs. In: Proc. of the 1998 Annual Computer Security Applications Conf. (ACSAC'98), Dec. 1998
  - 15 Anderson J P. Computer security threat monitoring and surveillance: [Technical Report]. James P. Anderson Co., Fort Washington, PA, April 1980
  - 16 Cannady J. Artificial neural networks for misuse detection. In: Proc. of the 1998 National Information Systems Security Conf. (NISSC'98). Arlington, VA. Oct. 1998. 443~456
  - 17 Porras P A, Kemmerer R A. Penetration state transition analysis - a rule-based intrusion detection approach. In Eighth Annual Computer Security Applications Conf. IEEE Computer Society Press. Nov. 1992. 220~229
  - 18 Schultz M G, Eskin E, Zadok E. Data mining methods for detection of new malicious executables. In: Proc. of IEEE Symposium on Security and Privacy (IEEE S&P-2001). Oakland, CA: May 2001
  - 19 Lee W, Stolfo S J, Chan P K. Real time data mining-based intrusion detection
  - 20 蒋建春, 马恒太, 任党恩, 卿斯汉. 网络安全入侵检测: 研究综述. 软件学报, 2000, 11(11): 1460~1466
  - 21 胡侃, 夏绍玮. 基于大型数据库的数据挖掘: 研究综述. 软件学报, 1998, 9(1): 53~63
  - 22 Han Jiawei, Kambe M. 数据挖掘概念与技术. 机械工业出版社, 2001