

# Internet 上的软件 Agent<sup>\*</sup>)

Software Agent on Internet

沈达阳 林作铨

(汕头大学计算机科学研究所 汕头 515063)

**摘要** 本文首先介绍 Agent 的基本概念及其相关理论、体系结构和语言, 然后较详细地介绍软件 Agent 方面的典型工作; 最后, 阐释 Internet 软件 Agent 有关思想, 并给出一些相应的研究成果。

**关键词** Agent, 软件 Agent, Internet | 软件 Agent

TP393  
TP317

## 1 引言

随着 Internet 的迅速发展, 各种类型的信息服务层出不穷, 许多技术问题的解决亟需智能软件技术的帮助。Internet 上的软件 Agent 可以十分灵活地提供多种智能化的信息处理手段, 将有力地开拓 Internet 上的信息服务, 使更多的人, 尤其是非计算机专业人员, 投入到 Internet 的发展洪流中来。另一方面, 作为一个软件环境, Internet 又是考验软件 Agent 技术的一个非常具有挑战性的现实和真实的平台。

目前关于 Agent 的研究, 无论是在 AI 界还是在计算机科学的其它领域, 都十分活跃。从 Agent 一词衍生出许多新的概念和相关研究领域, 从哲学、理论到技术, 对 Agent 一词的概念、理解和使用因所研究和关心的问题不同很不一致。近来有关 Agent 的研究热点主要是指所谓的软件 Agent (姑且译为“软件人”, 特别指相对于机器人而言)。即使是软件 Agent 现在也是各执一词, 例如有 knowbot、softbot、userbot 这些名称。软件 Agent 是对建立软件人这一新的研究领域的统称, 作为 Agent 研究的一种重要方法, 它引起了不同领域研究人员的广泛注意。然而, 它更加深入地包含了人工智能研究的许多方面, 因此是人工智能研究所希望的一个新进展。

## 2 什么是 Agent

“Agent”一词目前已被广泛引用, 但由于内涵极

其丰富, 目前尚无被普遍接受的简单定义。Wooldrige 在文[1]中给出了 Agent 的几种定义:

**弱定义** Agent 最一般地用以说明一个软硬件系统, 它具有这样的特性: ①自主能力 (autonomy): Agent 可以在没有人或其它 Agent 直接干预的情况下运作, 而且对自己的行为和内部状态有某种控制能力。②社交能力 (socialability): Agent 和其它 Agent (也可能是人) 通过某种 Agent 交流语言进行交互。③反应能力 (reactivity): Agent 观察其环境 (可能是物理世界, 图形界面, 一系列其它 Agent、Internet 等), 并在一定时间内做出反应, 以改变该环境。④预动能力 (pre-activeness): Agent 不仅简单地对其环境做出反应, 也能够通过接受某些启动信息, 体现出目标定向的行为来。

**强定义** 一些人, 特别是来自 AI 界的研究人员, 一般认为, Agent 不仅应该具有以上定义的特性, 同时也应具有一些通常人类才具有的概念, 如: 知识、信念、目的、义务等。Shoham 认为<sup>[9]</sup>, Agent 就是一种实体, 它可以被看成为由许多心智状态, 如信念、能力、选择和承诺等所组成的。一些 AI 研究人员更进一步考虑了 Agent 的情绪化。另外, Agent 还具有一些其它特性, 如: ①移动性 (mobility): 指 Agent 可以在信息网络上移动。②真实性 (veracity): 假设 Agent 不得传输错误信息。③仁慈性 (benevolence): 假设 Agent 没有冲突的目标, 因此每个 Agent 通常有求必应。④合理性 (rationality): 假设 Agent 总是为达到目标而努力, 而不阻碍目标的获得, 至少在它

\* ) 国家自然科学基金、广东省自然科学基金和汕头大学自然科学基金资助项目。

的信念中,应该如此。

实际上,对 Agent 还可以给出更多性质,但是至少目前还不能给出一般性准则,正如希望给出 AI 一般性准则一样,只是 AI 实践的继续<sup>[2]</sup>。有关 Agent 的研究主要包括三个方面<sup>[1]</sup>:理论、体系结构和语言。

### 2.1 Agent 理论

Agent 理论是指一些和 Agent 有关的基本规范。Agent 理论家研究这样一些问题:如何对 Agent 这个概念下定义,Agent 应具有什么特性,对这些特性,形式上应如何描述和推理等。

为什么对 Agent 来说,信念、愿望等概念是重要的呢?McCathy 认为:“当 Agent 的信念、愿望、目的、意识、能力、需求等特性,表示给计算机的信息和表示给人的信息是一样时,把它们赋予给计算机是合理的,因为它们能够帮助我们理解计算机的结构,预测它过去或将来的行为,或者提出修改和改进它的方法。”这对系统行为难于预测的复杂系统,是十分重要的。

由于经典的逻辑框架无法表示目的、信念等意识化的概念,Agent 理论家提出了一些新的形式化系统,以期从语法和语义两方面作出改进。

在语法方面,有两种基本方法,一种是模态语言,即把非真值功能的模态算子应用到公式中,另一种是采用元语言的方法。这方面的工作有:Levesque 提出的信念和意识理论(Belief & Awareness),Konloige 的演绎模型理论(Deduction model),Moore 的知识和行为逻辑(Knowledge & Action)等。其中比较著名的是 Cohen 和 Levesque 提出的 Intention (意图)理论。他们认为,一个 Intention 系统应该满足以下 7 个特性:

- Intention 向 Agent 提出问题,而 Agent 需要决定解决它们的方法。
- Intention 提供选择其它 Intention 的过滤器。
- Agent 能跟踪它们的 Intention 成功的踪迹,而且,一旦它们的 Intention 失败,能重试。
- Agent 不相信它们自己不会带来 Intention。
- 在某种环境下,Agent 相信自己会带来 Intention。
- Agent 不需要意识到它们的 Intention 的所有可能的后果。

基于以上的特性,他们构造了一种理性 Agent

逻辑(logic of rational agency),并在这个框架之下,引入了一系列派生的构造,Intention 是其中的一种。

关于 Agent 的语义方面,大部分工作都来自可能世界语义。在可能世界语义中,一个 Agent 的信念、知识、目标等都被描绘成一系列可能世界,它们之间有某种可达关系,可能世界语义可以和一致性理论相结合,使之成为一种引人注目的数学工具。但是,它也有许多相关的困难,如著名的逻辑全知者(Logical omniscience)问题。

实际上,有关 Agent 的理论研究在 AI 知识表示中已有较长的历史,特别是前些年提出的关于心智状态(mental states)的推理和关于行动的推理等研究是关于 Agent 研究的重要起点。

### 2.2 Agent 体系结构

在计算机科学中,体系结构指功能系统中不同层次结构的抽象描述,它们和系统不同的实现层次相对应。Agent 的体系结构也主要描述 Agent 从抽象规范到具体实现的过程。这方面的工作包括:如何构造计算机系统以满足 Agent 理论家所提出的各种特性,什么软硬件结构比较合适,如何合理划分 Agent 的目标等。Agent 的体系结构一般分为两种:

①主动式体系(Deliberative Architectures):根据经典的物理符号假设,包含显式表示的符号世界模型,并从中通过模式匹配和符号处理进行逻辑推理。在这个领域里,某种形式的规划系统,长期以来,一直被认为是智能 Agent 的核心部件。著名的工作有:Nilsson 等的 STRIPS 系统,Bratman,Israel 和 Pollack 的 IRMA 系统等。

②反应式体系(Reactive Architectures),即不包括任何符号世界模型,不使用复杂的符号推理的体系。这方面的工作有 Brooks 的 subsumption 体系,它由处于不同阶段的目标定向行为组成,更低层的行为更加原始(如避开障碍),但优先级更高。根据该体系构造的机器人,在没有符号 AI 系统中的显式推理和计算量相当简单的情况下,能够完成符号 AI 系统中一些引人注目的任务。这方面的工作还有 Pattie Maes 的 Agent 网络体系等。

许多研究人员认为,完全的主动式或反应式体系都不适合构造复杂的 Agent,其中提出了一些混合型体系,以期结合两者的优点,其中比较著名的系统有:Georgeff Lansky 的 PRS (Procedure Reasoning System)体系。PRS 是一种信息-愿望-目标体系,

它有一个计划库,以及显式表示的信念、愿望和目标。信息是一些事实,包括外部世界和 Agent 的内部状态。它们是用经典一阶逻辑表示的。愿望由系统行为来表示。PRS 计划库包含一些不完整的计划,称为知识区(KA),一个 KA 都和一个激活状态相连,这个状态决定 KA 什么时候被激活。KA 可以用目标,也可以用数据来激活,所以 KA 可以是主动式的,也可以是反应式的。Agent 中一系列活动的 KA 代表其目标。所有这些数据结构,用一个系统解释器来处理,它负责更新信念,激活 KA 或执行系统行为。这方面的工作还有:Ferguson 的 TOURING-MACHINES,Brumeister 等的 COSY 等。

Slovan 通过考虑人脑的信息处理体系,进一步认为<sup>[1]</sup>,一个智能 Agent 应该包含以下三个子系统:

- 反应式子系统:通过外部传感器和内部监控系统来获得信息,并在系统内部传播,同时通过效应器对外部产生作用。人和其他动物的学习能力都是反应式子系统的扩展。

- 主动式子系统:由于反应式子系统的功能往往限定于某种固定类型的信息,需要有一个新的子系统,它可以根据不同的信息来源,创建、修改、比较、评价和选择新的子结构,而这些子结构都是更原始的反应式子系统。该系统体现人类高层的规划、比较等能力。

- 元管理子系统:这是一个内部状态监控系统,它能够跟踪主动式子系统的系统进程,产生的问题和决定等,并执行评价,从而逐步改进系统的性能。它体现了人类高层的综合、学习能力。

此外,特别值得一提的是,具有知识的 Agent 的体系结构很大程度上和基于知识的系统结构相关。

### 2.3 Agent 语言

Agent 语言就是遵循 Agent 理论家提出的各种基本原则的程序语言,包括:如何对 Agent 编程,其基本单元是什么,如何有效地编译和执行 Agent 程序等。Agent 语言,不但用于构造与 Agent 相关的系统结构,也包含了 Agent 一些较强的特性,如信念、目标等。Shoham 最先提出的“基于社会性计算的编程体系”,其关键思想就是面向 Agent 编程(AOP)<sup>[9]</sup>。

AOP 可以看成是 OOP 的一种 Agent 化的扩展。和 OOP 中的类相似,AOP 中的模块(即 Agent),也通过不同的方式处理输入信息来和其他

模块交流,但 AOP 还提供了一种具有明确定义的语言框架,用以表达信念、能力、决定等成分。Agent 的精神状态,被加上各种限制,从而保证 Agent 的行为符合常识。Agent 的计算包括:通知、请求、提供服务、接受服务、拒绝服务、竞争、合作等,这些都借鉴了言语行为(speech act)理论中的一些概念,以表达同一言语行为,在不同条件下具有不同效果。Shoham 实现的 AOP 系统称为 AGENT-0,它包含三个部分:①一个决定 Agent 精神状态的逻辑系统。②一种用于 Agent 编程的解释性程序语言。③一个 Agent 化的进程,把 Agent 程序编译成低层的可执行系统。

McCarthy 的 Elephant 2000<sup>[10]</sup>中的 Elephant 语言实际上是 AOP 的先驱,它主要借鉴言语行为理论中的言外之意(illocutionary act)和言后之意(perlocutionary act)两个概念,使该语言显得更接近于自然语言。

- Elephant 中的 Agent 互相交流所采用的语言,具有明确语义的言语行为。

- 程序的正确性部分地定义为言语行为的正确执行,即 Agent 的回答应该具有真实性和相关性,要遵守诺言。

- Elephant 源程序不需要数据结构,但它可以直接引用过去事件,因此它和一般的程序相比,更接近自然语言。

- Elephant 程序可以用逻辑语句表达,它的输入输出规范和执行规范分别是言语行为理论中的言外之意和言后之意的推广。

目前,Elephant 2000 的工作只是一个草图,但是其特点很明显,那就是强调 AOP 的行为性(performative),即 AOP 中的许多句子不仅具有说明性的意义,而且主要是一种行为。此外,如 Reiter 和 Levesque 等人提出的 GOLOG 语言也可以看成一种 Agent 语言。

随着 Agent 技术的成熟,用于设计基于 Agent 系统的软件工具也日益增多。如:General Magic 公司的 Telescript 就是商品化的 Agent 语言。Telescript 的开发对象是电子化商场。在 Telescript 中,有两个关键概念:Place 和 Agent。Place 是 Agent 的可占据的虚拟位置,Agent 是电子化商场中的供应者和消费者,它是用软件进程来实现的,可以和其他 Agent 交流,同时也可以移动。

和 Agent 相关的工作有很多,早期的工作大部分来自 DAI 和 Robotics 等领域,随着 Agent 技术的发展,在其它许多计算机科学领域也得到广泛的应用。软件 Agent 就是其中一个较好的发展方向。

### 3 什么是软件 Agent

和 Agent 一词相似,软件 Agent 这个概念,也因研究者不同而存在着差异,也有广义和狭义之分。狭义上,软件 Agent,是指活动于软件环境中的 Agent 软件系统,它通过下达命令和分析环境反馈同环境进行交互。Etzioni 认为利用软件环境来进行 AI 研究,特别是 Agent 研究,和硬件环境相比,有以下优点<sup>[3,4]</sup>:

- 方便实用:软件开发和系统化实验所需的费用、精力和专门知识相对较少,而且软件 Agent 的研究成果比较容易传播和复制。

- 真实丰富:和模拟布置的物理环境相比,软件环境是真实的,同时也能提供各种丰富的,类似物理环境中所具有的测试例子,如:直觉、启发性实例、令人迷惑的阻塞等。

- 研究内容集中:软件环境可以排除一些棘手问题的发生(如:克服传感器噪音,对液体、形状的表示等),从而能够集中研究核心 AI 问题。

一般地说,软件 Agent 应该具有以下特点:

- 目标定向的行为:软件 Agent 试图达到一定明确的目标,人可以通过对软件 Agent 的“编程”来确定其目标。

- 规划、执行和错误修复能力:软件 Agent 能执行一系列行为(它所知的,或由它自己根据目标生成的),一旦执行,就可达到其目标,而且软件 Agent 在执行命令序列时,能够监视其行为,并试图修复任何预料不到的错误或失败。

- 陈述性知识表示:软件 Agent 采用陈述式的知识表示方法,能够在不同任务中运用同一知识。

- 学习和适应:随着时间推移,软件 Agent 通过内部记录,归纳经验,逐步改变其自身的性能。

- 连续性操作:软件 Agent 在环境中可以连续操作,因此其设计者应该考虑其生存和共存等问题。

- 自然语言处理:在软件环境中,由于软件 Agent 可获得的大部分信息资源,都是以自然语言的形式出现的,因此软件 Agent 对其环境的理解,随其自然语言理解能力的提高而提高。

- 通讯和协作:在软件 Agent 的环境中,可能存在多个其它的 Agent,故软件 Agent 达到目标的能力与其通讯和协作能力相关。

- 移动和复制:不象其它软件,软件 Agent 是移动的,可以通过自动复制,从一个计算机移动到另外的计算机。

Etzioni 的 Softbot 是一种通过命令和对软件环境反馈的解释来与软件环境交互的软件 Agent。Softbot 的效应器,就是用来改变外部环境的命令(如 UNIX 中的 Shell 命令 mv 或 compress)。Softbot 的传感器,就是用来提供环境信息的命令(如 UNIX 中的 Shell 命令 pwd 或 ls)。通过 Softbot,Etzioni 为操作系统的命令界面提供了一种面向目标的方法(称为 OS Agent)<sup>[5]</sup>。

在 OS Agent 中,首先由用户简单地指定需要完成任务,然后由 Softbot 利用系统的状态和系统命令所获取的信息,决定如何完成该目标。Softbot 动态地合成相应的命令序列,产生所需的命令和系统调用,并处理系统错误,在必要的情况下,还可以多次尝试同一命令,直到完成目标。通过 Softbot,可以把 AI 中规划和学习等技术应用到操作系统环境,从而增加了用户命令的能力。目前,OS Agent 可以处理的内容有:①事件:在本周内,一旦电子公告牌上出现包含“自行车”的公告,就把该公告用电子邮件发给我。②限制:使我的议事日程和计算机系的访问日程保持一致。③脚本:每天早晨,从名叫“June”的机器取来我的电子邮件,然后启动我的新闻阅读程序,显示当天的天气预报。④集合指定:压缩目录中六个月以上没有修改过的 Postscript 格式的文件。

和操作系统相比,Internet 这一软件环境,显得更加复杂,更具有挑战性,因此也更需要 Agent 技术。Internet 软件 Agent,就是该技术在这一领域的应用。

### 4 Internet 上的软件 Agent

Internet 上的软件 Agent,就是以 Internet 这一规模庞大、极度异质、高度动态的软件环境为研究基础以提供对 Internet 信息的收集、检索、分析、综合等高度体现智能行为的信息处理手段为目的的软件 Agent。它既可以充分应用现有的人工智能技术成果,开拓 Internet 上的应用,又可以利用 Internet 这

样一个非常具有挑战性的平台,检验人工智能理论和发展人工智能研究,因而 Internet 软件 Agent 是一个十分诱人的新领域。

目前,国内外的许多机构和企业已经为 Internet 提供了用于创建、浏览、访问、搜索、阅读、交流信息等形形色色的服务,这些服务涉及了广泛的信息类型,包括:文本、声音、图形、图象、实时视象以及多媒体合成信息,其中 Yahoo 和 Infoseek 等信息搜索服务已经广为人知。但是,这些服务的功能都比较低层和原始,缺乏智能的处理能力,因而加重了最终用户的负担,使 Internet 的使用往往仅局限于计算机专业人员,严重地影响着 Internet 的进一步推广应用,即使是专业人员,在 Internet 上进行富有魅力的冲浪也正成为负担。另外,随着 Internet 的发展,各种 Internet 服务的集成变得越来越重要,如 Net Communication 公司的浏览器 Netscape 就已经集成了 http、ftp、gopher 等多种服务,但是,这些集成同样停留在原始的、功能性的阶段,能体现智能行为的集成方法尚未出现。另一方面,许多人盼望能有自然语言的 Internet 交互界面,可是,这种技术即使是在自然语言理解的传统研究领域中都难以指望在短期内实用。总之,由于缺乏智能化的信息处理手段,Internet 这一庞大的信息库的潜能远远得不到发挥。

早期的 Internet 上的软件 Agent,也称 Webbot, Webcrawler, Spider, WWW Worm 等,能够持续地运用于 Web 上,从一个页面跳到另一个页面,收集 Web 本身的统计信息或建立检索 Web 内容的集中化数据库。象 Alta Vista, Lycos, OpenText, WebCrawler 等一些流行的 Internet 查询网站都用 Spider 来帮助建立 Web 索引。不过,目前这些 Spider 的智能程度都比较差,对 Homepage 的语法和语义处理都还缺乏相应的理论和技术,因此,提供 Internet 信息查询的企业还需雇佣专门的 Internet 冲浪员来对 Spider 找到的信息作进一步的处理。目前,Internet 软件 Agent 的研究成果还比较少,主要有以下几方面的工作。

#### 4.1 信息交互、日程安排等工作

Katia P. Sycara 和 Dajun Zeng 设计的 Vistor-Hoster 来访安排系统包含两种类型的软件 Agent<sup>[4]</sup>,一是信息 Agent,如 Finger Agent 用来处理通过 Finger 命令获得的信息、Who's-Who Agent 用来处理通过 HTTP 协议提供的 Who's-Who 信息、

Faculty Interest Agent 获取大学职员研究兴趣等;二是任务 Agent,如 Vistor-Hoster Agent 负责接受来访者的信息、Scheduling Agent 提供日程安排和解决日程冲突、Personnel Finder Agent 提供和合并所查询的职员信息等。

Vistor-Hoster 可以完成这样的工作:假设 Minsky 想要访问 CMU 的计算机系,而且想和对机器学习有兴趣的研究人员会面,那么,Vistor-Hoster 就请 Minsky 输入他有关的信息(如:机构名称、e-mail 地址等)、访问日期、以及他对与会者的要求(即机器学习方面的研究人员),然后, Faculty Interest Agent 取来符合 Minsky 要求的潜在与会者名单,接着 Personnel Finder Agent 被激活,同时发出大量的查询信息,通过各种信息 Agent,如 Finger Agent, Who's-Who Agent 收集个人信息, Personnel Agent 对信息进行合并,解决其中的冲突。Vistor-Hoster 从中选择一些老资格研究人员,自动发出 e-mail,询问是否愿意在 Minsky 指定的日期与之会面,一旦收到同意的回信,Vistor-Hoster 启动 Scheduling Agent,在与会者个人的 Scheduling Agent 的帮助下(即合并个人日程信息,解决冲突),生成可行的日程表,提交给 Minsky 和所有的与会者。

#### 4.2 Internet 的浏览助理

Balabnovi 和 Shoham 等构造的适应性 Web 自动浏览 Agent<sup>[5]</sup>,能够帮助用户获得感兴趣的最新信息。它的工作是:每天提供用户感兴趣的 Web 页面的选集,得到用户的反馈后,系统作出调整,并在以后提供更好的页面。

该 Agent 把 Internet 看成是一系列可以通过统一资源定位标记(URL)存取的页面。采用最简单的搜索技术:best-first 策略,并用资源限制来约束每一个循环 Agent 所接触的页面。在特定的时间限制内,使用不连续的循环,对 web 进行搜索。其步骤如下:

- 使用某种启发式选择方法(可能和搜索所采用的启发式信息相同),选择最佳的 p 个页面提交给用户。
- 从用户那里接受每一个页面的评价。
- 根据用户的反馈,更新搜索和选择的启发式信息。

#### 4.3 信息检索或收集

由于 Internet 是一个高度异质的信息环境,和自然环境有一定的相似性。Menczer 等把人工生命方面的技术应用到 WWW 的信息检索<sup>[4]</sup>。他们构造了许多能在 WWW 上运行和抽取数据的软件 Agent,通过稳定的遗传算法,使 Agent 的数量逐步进化。Agent 的存活、复制和死亡都和系统能提供的能量有关。Agent 在每一个系统循环中,从环境和内部状态中接受信息,进行计算和执行各种活动,而活动需要能量的输入。任务完成得好的 Agent 复制得越多,不好的 Agent 逐步被淘汰。Agent 在有限资源的环境中进行竞争,同时,Agent 的基因变异提供了动态环境中 Agent 进化所需的变化。这种体系提供了一种和信息密度相关的自然选择的过程。Menczer 把这种体系运用到 WWW 的信息检索,使提供检索任务的 Agent 集中活动在 WWW 上用户关系的内容较多而且存取效率较高的数据区。

我们目前所构造的 Internet 信息收集 Agent 也是一种 Internet 的软件 Agent。它是一种具有知识(常识)的软件 Agent,并用 Java 编程,可以充分利用 Java 等现有软件工具所提供的包括并发处理等性能,其任务也是收集 Internet 上用户感兴趣的信息。但和前者不同,它采用基于关键词匹配和对 Homepage 语法信息制导的方式进行搜索,同时根据国内 Internet 的运行特点(如连接国外网点的数据传输率较低,网络联接容易中断等),提供后台搜索、中断再联等功能。它的一种最简单的系统由三个层次的 Agent 组成:

- 最高层的 Agent 是 Monitor,主要负责系统的监控工作,如系统资源是否耗尽,是否有空闲的 Agent 等。

- 中层的 Agent 是 Collector,主要负责系统的搜索工作,系统可以同时运行若干个 Collector,在不同的网点进行搜索。

- 底层的 Agent 是 Analyser 和 Filesaver,分别负责系统的低层文件分析和文件存储。

目前该系统运行在 Sun-Solaris 之上。我们将在系统中逐步建立收集系统,Homepage 理解系统,Internet 知识库和推理机系统和自然语言处理以及提供信息收集的交互语言等。应该指出,软件 Agent 研究,特别是 Internet 软件 Agent,必须以实现系统为

目标,给出包括用户界面的 Internet 软件,但同时必须解决关键的理论问题,特别是软件 Agent 的规范和用户与 Agent 的交互语言的描述形式。

**结束语** 目前,人工智能研究的许多方面都已经硕果累累,而 Internet 作为一个全球数字化信息库,其对人类社会日益广泛的影响也是不可阻挡的。Internet 上的软件 Agent,不但可以充分应用现有人工智能理论并深入发展人工智能研究,而且可以得到具有广泛应用前景的研究成果,故具有更大的研究和应用价值。可是,有关 Internet 和软件 Agent 的研究也容易形成一种不成熟的局面,那就象早期对专家系统的研究热潮,一拥而上,对软件 Agent 的研究必须有相应的基础研究作为基础,只有这样才能走出一条自己的路子。

#### 参考文献

- [1] M. Wooldridge 等, Intelligent Agents: Theory and Practice, Knowledge Engineering Review, 10(2)1995
- [2] John McCarthy, 人工智能需要更高标准, 林作铨译, 计算机科学, 1994
- [3] O. Etzioni 等, Building Softbots for UNIX, Preliminary Report, Washington University
- [4] O. Etzioni, Intelligent Without Robots (A Reply to Brooks), AI Magazine, Dec., 1993
- [5] O. Etzioni 等, OS Agents: Using AI Techniques in Operating System Environment, Technical Report 93-4-04, Washington University
- [6] Kasia P. Sycara 等, VISITOR-HOSTOR: Towards An Intelligent Electronic Secretary, CIKM'94, Maryland, USA
- [7] M. Balabanovic 等, An Adaptive Agent for Automatic Web Browsing, Working paper SIDL-WP-1995-0023, Stanford University
- [8] F. Menczer, 等, An Endogenous Fitness Paradigm for Adaptive Information Agents, CIKM'94, Maryland, USA
- [9] Shoham, Agent-Oriented Programming, AI, 60(1)
- [10] John McCarthy, Elephant 2000: A programming language based on speech acts, unpublished paper, Stanford University
- [11] A. Sloman, What sort of architecture is required for a human-like agent?, Invited talk at Cognitive Modeling Workshop, AAAI96, Portland Oregon, Aug. 1996