

Semantic Web、Agent 和网络虚拟社会

戴欣 申瑞民 张同珍

(上海交通大学计算机科学与工程系 上海 200030)

Semantic Web, Agent and Network-Virtual Society

DAI Xin SHEN Rui-Min ZHANG Tong-Zhen

(Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200030)

Abstract This paper tries to discuss one realizable mode of SW(Semantic Web). It is called NVS(Network-Virtual Society). SW is regarded as the next-generation Web. By adding semantics into Web, SW provides interoperability between applications and facilities to enable automated processing of Web resources. Agent will be the executer in the automated process. After analyzing relational theories and technologies, we put forward the concept and mode of NVS, and gives our reason.

Keywords Semantic Web, Agent, Network-virtual society, Ontology

1. 引言

最初的计算机是用于计算的设备。到了 80 年代, PC 发展成多功能(如: 游戏, 文本处理和 PowerPoint 呈现等)的系统。现在计算机变成了接入互联网的入口。互联网技术正飞速发展, 使人们足不出户就能处理很多和外界交互的事务。网络席卷全球, 成为人们生活不可或缺的一部分。美国网址专家也是网址杂志的创办人凯利说: 自网景公司于 6 年前申请上市以来的 2000 天中, 人类居然创写了 30 亿网页, 建立了 2000 万个网址, 而传送的电子邮件就达 3 兆 5 亿则之多, 真是不可思议。

然而, 目前的 Web 技术存在影响其发展的一些严重障碍。初期的一些简化导致制约信息查询、提取、保持和生成的瓶颈^[3]。计算机仍是信息传接、呈现的设备, 不能真正涉及到内容。这样, 计算机只能对信息的访问和处理提供有限的支持, 大量信息提取和解释工作还需要人工完成。

互联网容量的持续快速增长使得信息查询、定位、组织、整合越来越困难。面对海量信息的处理, 我们需要将部分任务交给机器处理。但由于自然语言理解的问题仍未解决, 机器不能很好地理解网页内容, 从而无法胜任我们希望交给它们的任务。一个替代方法是对 Web 进行改造, 使其更容易被机器理解。Semantic Web 应运而生^[3]。

2. 语义网(Semantic Web)

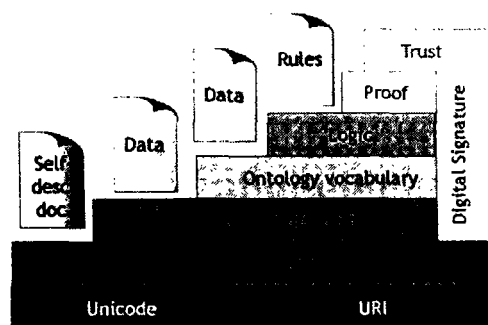


图 1 Semantic Web 的层次图

Semantic Web 的概念是 Tim Berners-Lee 在 XML 2000 会议上首先提出的。他认为 Semantic 意味着是机器可处理的, 并能用语义测试予以检测: 如果你将数据交给机器, 机器能根据语义做出正确的事^[8]。Berners-Lee 所描绘的 Semantic Web 框架为层次结构(图 1)^[18]。

2.1 XML+XML 模式

XML(eXtensible Markup Language)是计算机可读文档的规范。它是 SGML 的一种约束应用形式^[15]。XML schema 的目的是提供 XML 标记库, 用来定义、描述和分类 XML 文档中的词汇^[16]。每一个 XML 文档都是有序的、标记了的树形结构。你可以编制各种语法清楚的数据结构。但 XML 不描述数据的使用和语义^[5, 6]。

2.2 RDF+RDFS 模式

RDF(Resource Description Framework)是元数据处理的基础。它为机器可理解信息的交换提供支持, 以实现 Web 上应用程序互操作性和 Web 上资源的自动处理。RDF 被用在多个应用领域, 如: 资源发现, 提高搜索引擎的性能; 分类, 描述具体网站、网页或数字图书馆的内容和相互关系; 方便智能 Agent 的知识共享和交换; 内容分级; 私有策略; 数字签名等^[17]。

RDF 的基本数据模型很简单: 一般包括资源、特性、声明^[3, 9]。还有文字, 它与资源一样具有实体性质。一个声明是一个三元组: Statement = Resource + Property + (Resource | literal)

看一个简单句子:

Ora Las is the creator of the resource

<http://www.w3.org/Home/Las>.

这个语句可分解成:

Subject (Resource): <http://www.w3.org/Home/Las>

Predicate (Property): Creator

Object (literal): "Ora Las"

可以用图示的方式表示上述关系, 见图 2。图中椭圆形的结点代表资源, 有标识的箭头代表特性, 矩形框代表文字(Literals)^[17]。

戴欣 硕士, 研究方向为 XML 技术, 智能代理。申瑞民 博导, 研究方向为数据挖掘, 远程教育。张同珍 讲师。

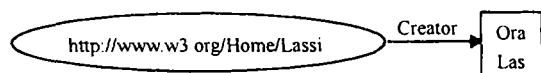


图2 简单的结点箭头图

从上例可以看出, RDF 并不关心句法, 它仅提供元数据表示的模型。除了图示方法表示外, 上述三元组也可以表示。当然, 还可以有其他方法。XML 显然是理想的候选者。RDF 数据模型规范包括基于 XML 的编码方式^[5]。

RDF Schema 是 RDF 的类型系统。它提供一种机制用来定义域中资源的类和特性, 以及这些特性如何作用在类上^[19,12]。

2.3 Ontology

Ontology 在牛津英文词典中的定义为: 事物的科学及研究 (the science or study of being)。在人工智能领域, 它被用作概念化的本质表示。也就是, 定义术语及其关系, 在正规的形式且最好是机器可读的方式下。James Hendler 认为, 在 Semantic Web 领域内, Ontology 是一套知识术语集, 包括词汇 (Vocabulary)、语义关联和一些简单的逻辑推导规则^[1]。

Ontology 中定义的概念在各个专门领域的实际运用——实用数据——与 Ontology 一起构成 Semantic Web 的基础。在新出的一些 Semantic Web 的试验原型中, 不同组织、不同背景、不同目标的试验者们创立了多种语言来表达 Ontology 和实用数据。典型的语言具有域内类的继承性描述、类的属性和类的实例等概念。现有的这些 Semantic Web 语言包括: SHOE、Topic Maps、XOL、RDF 和 RDFS 和 DAML + OIL^[4,20]。

从某种意义上说, Semantic Web 的生命力和发展性取决于 Ontology 的成熟度。许多组织和学术机构在做这方面的研究。Natalya F. Noy 等人的 Protégé-2000 工具用于 Ontology 的扩展和知识获取, 能兼容不同 Semantic Web 语言的编辑模式, 比较方便^[20]。Alexander Maedche 和 Steffen Staab 提出的 Ontology Learning 框架拓展了常用 Ontology Engineering, 用半自动的 Ontology-construction 工具实现^[7]。Jeff Heflin 的 Exposé 是一个蠕虫工具, 用来查找使用 SHOE 标记的网页用来扩充和验证所选 Ontology^[3]。Dieter Fensel 等人更加推崇 OIL, 研究并认为 OIL 工具包括三方面: Ontology 编辑器、标记工具和对 Ontology 的推理^[9]。

2.4 Logic——推导引擎

Logic 是构建推导引擎的理论基础。通过推导引擎从已有知识中推出新的知识。不同的逻辑方法在知识表达和推导方面有不同特性:

◇高阶逻辑 在所有逻辑中, 高阶逻辑具有最强的表达能力。如: 关系的传递闭包特征需要用高阶逻辑描述。然而, 没有很好的可计算性。还存在不可证明的语句 (Goedels 的不完备性理论)。

已经证明, 如果不需要语义的高阶特性的话, 二阶逻辑可转化成一阶逻辑。Michael Kiefer 的 Frame-logic 就是高阶逻辑下的一阶语义。

现有的基于高阶逻辑的推导引擎如常用定理的证明器 (Isabelle、HOL)^[21]。

◇一阶逻辑 是半可决定性的, 大规模数据和公理下计算起来会难以驾驭。"Thousands of Problems for Theorem Pover (TPTP)" 中列出了许多小的但暗藏机智的问题是自动

定理证明所难以解决的^[22]。

这意味着网络环境下面对大量的知识、数据, 单纯一阶逻辑方法实现起来很困难, 所以往往一些一阶逻辑的子集有较好的性质。如: SETHEO。

◇描述逻辑 (DL) 允许用一套约束的一阶公式说明继承性。它们有令人满意的计算性质 (可决定性、易控性、较好的平均计算时间)。但推导功能局限在分类和包含。这意味着基于 DL 的推导工具仅能判断某个公式所描述的类是否是其他类的继承, 或者某个实例是否是类的实例。

另一个可行的 Semantic Web 推导引擎是基于 Horn-logic 的。

◇DataLog 是 Horn-logic 的一个子集, 被用在推演数据库、逻辑语言和一些有效评估策略方面的研究。自顶向下是 Prolog 语言的推导方式, 但碰到描述含环的传递性推导时会出现问题。Courteous logic programs 通过引入优先级化的冲突处理来解决问题^[23]。其应用实例有 SiLRI, 一个完全基于 Java 的系统。

2.5 Prove 和 Trust

随着其底层技术的不断完善, Semantic Web 应用的推广、信用评估和安全问题将会日益突出。

3. Agent

从某种意义上说, Semantic Web 概念的提出, 相当部分是建立在 Agent 技术成果之上的。多 Agent 系统间通信的许多挑战都需要 Ontology 的研究突破。Agent 技术和 Ontology 的结合将极大地推动网络服务的发展, 增强程序自动处理的能力, 减少人为干预^[1]。

3.1 Agent 的概念

一个 Agent 弱的概念。Agent 通常被理解成一个软硬件结合的计算系统, 具有如下特性:

◇自治: Agent 的操作不需要其他 Agent 或人为直接干预, 有某种程度的行为控制和内部状态。

◇社交能力: 能用 Agent 交流语言与人或其他 Agent 交流。

◇反应: Agent 能感受环境, 并作出及时的反馈, 调整状态。

◇以我为主: Agent 不仅对环境作出简单反馈, 而且主动为完成目标采取相应行为。

简单说来, Agent 可以理解为具有上述特性的象 UNIX 中的进程。

一个 Agent 强的概念。一些 AI 的研究人员认为 Agent 是一个计算机系统, 除了具备上述特性外, 还应有一些人的特性。例如, AI 领域认为 Agent 具有心灵特性, 包括: 知识、信仰、意图、责任; 有的走得更远, 甚至包括情感^[24]。

3.2 Agent 技术概要

Agent 技术区别于传统技术的主要特征是, 它所具有的自制能力、智能和目标驱动属性, 能够通过各种社交、学习、推理等方法感知和适应复杂的动态环境, 具有自动追求目标的能力。Agent 技术是传统对象技术的发展和飞跃, 具有强大的开放性、灵活性、适应性和协作性。然而, AOP 程序自身具有信念的同时, 还具有意愿、期望和知识等。AOP 中各 Agent 的行为是在其信念的指导下, 受其本身知识以及由此产生的期望和意愿的支配。AOP 程序的执行过程是各 Agent 之间相互协商与协作的过程, 具有一定的不确定性, 其问题求解的过

程是设计人员和 Agent 系统共同努力的结果^[27]。

逻辑是 Agent 智能的基础。前面各种逻辑介绍的工具都可看成低级智能体。许多 Agent 语言都是在经典逻辑和新推出的逻辑的基础上通过引入算子、增加限制等手段形式化地建立起来的。当然,新提出的语言往往加上 Web 特性。Anton Eliëns 等人的 WASP(Web Agent Support Program)项目关注的是建立 Web Agent 以帮助一般使用者跟踪网上信息^[25]。

面向 Agent 的方法也越来越多地用在软件工程中^[24]。

如前所述,Ontology 是多 Agent 通信所研究的关键技术之一。我们知道 Agent 间的交流需要一套经得起考验的知识状态、动作定义、规划的表示系统。逻辑符号是最好的表示方法。将逻辑加入 XML/RDF/DAML 中对基于 Web 的 Agent 行使其功能来说是一个很好的主意。除表达能力外,逻辑表达式的计算复杂性问题通过好的算法解决。规划算法很自然地适合于服务描述的思想。服务描述实际上是 Agent 各种接口的组合,一个规划就是为了达到某个目标设计的接口的顺序。规划或多或少由规划算法完成。然而,规划算法还应能多方面拓展,以便处理 Agent 知道的和目标 Agent 想得到的之间的不同。出现这样的情况主要有两方面的原因:Ontology 不匹配和数据结构不匹配。前者需要词汇和正式推导过程的完善,后者则要能做到自动数据结构的转化^[14]。DAML(DARPA Agent Markup Language)是美国国防部研究机构 DARPA 为实现 Semantic Web 而开发的语言。DAML+OIL 是一套语义标记语言,可以方便地用于定义 Ontology,并对 Agent 提供良好的支持^[1,11,26]。

4. 网络虚拟社会

多数专家都认为 Agent 是 Semantic Web 的一部分。在不久的将来,机器将能聪明地进行交谈,帮助你我解决各种复杂问题。技术人员在为下一代网络——Semantic Web 的实现而不懈努力。没有人能肯定 Semantic Web 会最终发展成多大。就像近几年人工智能和软件 Agent 一样,夸大的宣传还是有的。但看起来,其潜力还是很大的,规划网络发展的人们不断将其概念复杂化。正如 W3C 主席 Jean-Francois Abramatic 所说,Web 已经提供了普遍信息访问的潜力。从大的方面来说,今天的 Web 信息只是用 HTML 语言显示在屏幕上。我们所要做的是设计方法帮助机器理解它们所显示的,这样它们就能彼此交流,共同协作,解决更大的问题。Semantic Web 的提出无疑让人们看到了希望,但它会发展成什么样子?什么才是它正确的发展道路?这是我们应该考虑的问题。

4.1 专家对 Semantic Web 的一些构想

Tim Berners-Lee, James Hendler 和 Ora Lassila 在美国科学杂志上撰文介绍了他们对 Semantic Web 美妙前景的预测^[10]。随着 Web 内容的机器可读和各种自动服务的增多,网上 Agent 的数量将呈几何级增长,很多事务可由各种 Agent 相互协作完成,Agent 还能通过相互“证明”消除相互间的不信任。Agent 还会做广告,宣传自己的功能。接下来,URI 能指向任何东西,包括实物。这时也许你将会看到,当你接电话时音响的声音自动收小。

斯坦福的 Sheila A. McIlraith 等人已经对 Semantic Web 的服务展开了研究^[2]。他们以 DAML 系列标记语言为基础,将 Web 服务分成三类:

- ◇自动 Web 服务发现。涉及到服务的定位和性质判断;
- ◇自动 Web 服务执行。包括程序或 Agent 自动执行特定

的 Web 服务;

◇自动 Web 服务组合和不断交互。自动对各个服务进行选择、组合,执行过程中不断交互,以完成复杂的任务。

他们在高级逻辑语言 ConGolog^[13]的基础上给出了一个 Agent 与 Web Services 互动的会议行程安排的实现框架。由 Agent 查询各个 Web Services 信息,结合日程要求安排交通、住宿、会议行程。

这些想法和工作都很有意义,但似乎总让人觉得缺乏总体方向性把握。

4.2 网络虚拟社会的概念

在此,我们想提出“虚拟网络社会”的概念。即网络模仿现有的社会运行模式运行。比如:想查一本新书是否有卖。我告诉我的 Agent Lucy,书名和作者。每个网上书店都有 Agent 负责接待、咨询。Lucy 逐个问他们:“你们书店有这样书名和作者的书吗?我叫 Lucy,有消息通知我。”Lucy 这样跑了一圈,那些 Agent 或自己或通知店里其他 Agent 查询是否有这本书,并将结果通知 Lucy。Lucy 综合以后告诉我。Lucy 也能帮我做上述会议行程安排。只不过,她这次联络的是各个航空公司、铁路公司订票咨询和酒店咨询的 Agent,并将收集的信息进行筛选,设计符合我的要求的行程方案。

也就是说,所谓网络虚拟社会,就是网络的人类社会化;网络社会中,Agent 相当于人,网站相当于一个组织、机构、部门、甚至家庭等社会实体;每个这样的社会实体由授权的 Agent 进行管理,体现区域自治性。复杂的问题可由多个 Agent 协作完成。Agent 负责管好自己的辖区,是自己领域内的专家,对自己不知道的、不了解的就请教别人。网络社会的运行是其中所有 Agent 合力的结果。

4.3 网络虚拟社会的合理性

首先,Agent 就是 AI 领域模仿人的特性进行研究的。那么把 Agent 看成网上人的化身就是很自然的事情。

其次,现在各行各业,各个组织部门都十分流行上网,例如:政府上网,公司上网,学校上网,个人上网……等等,这不就是把原有社会实体的职能搬到了网上吗?随之而来的电子政务,电子商务,网络教育,个人主页……,不就是将原来的各项社会活动,尽量放到网上进行吗?

由此不难看出已经存在的对网络虚拟社会的需求。

从人性的角度看,人类的本位主义倾向是不言而喻的。他们按照自身的情况构造了神的世界;他们按照自己所期望的生活方式改造着世界。现在他们致力于进一步培养和创造机器,以期按照他们的生活模式构造一个机器世界。人工智能本身的研究就是这一心理潜向的最好注脚。

把 Semantic Web 发展成网络虚拟社会是符合人类这种心理潜质的,而这种人性的力量是无比巨大的。

4.4 向网络虚拟社会发展的技术可行性

先谈谈 Semantic Web 发展目前存在的最大问题: Semantic Web Ontologies。

在 James Helder 看来, Semantic Web 的发展不能只是包括某一种单纯的(neat)Ontology,哪怕它是 AI 专家多么精心设计的结果^[1]。原因很简单,如果是这样,它就限制了 Semantic Web 的拓展性,而这是与 Internet 兼收并蓄的内在精神冲突的。反过来,要是没有这样的 Ontology,所有机器间完整的相互交流、相互理解就很难实现。这里似乎存在自身的悖论。Helder 预想,与 Web 的其他方面一样,Web 加入语义也呈无政府主义状态的复杂性。他谈到,在过去的几年,几乎所

有公司、大学、政府机构都想将自己的网页与 Ontology 建立关联,因为这样就可以运用多种有力的工具更最大限度地发挥那些内容的作用。在此之上,基于 Agent 的计算将更可行。Semantic Web 的美好前景似乎就要实现。

然而,一种 Web 发展初期的现象似乎又重现了:网页的所有者在看到其经济价值之后,才有动力用 Semantic 标注网页;能体现这一价值的工具往往又是在网页被标注之后才会被开发。尽管这一鸡和蛋的问题已被 DARPA 注意到,DARPA 也资助研究者在两方面同时前进,试图做最初的推动,但是没有一个杀手级的应用(killer apps)或解决方案显示 Semantic Web 的威力,仍会有很长的路要走^[1]。

简单地讲,要设计一个 Agent,能读懂各种 Ontology 标注的网页信息或者能与用各种不同的 Ontology 武装起来的 Agent 交流,是一件十分困难的事。而这一点不实现,Semantic Web 的价值无从体现。大家就只好继续观望。而从 Internet 全世界热火朝天参与下发展神话的经验来看,大多数人的观望是低效的。

网络虚拟社会的架构能通过分级或者说封装将此难解的问题分解,降低难度。简单地讲,就是将问题分成两级。一、每个 Web Site 由自己的 Agent 负责管理,实现区域自治。他是专门为本地 Ontology 设计的,他对自己所辖范围十分了解,能根据语义信息快速准确地得出正确结论。二、网络上不同区域的 Agent 采用统一的 Ontology 进行交流,并将其应用到 Internet 上。就像前面介绍的 Lucy 一样,她将进行自己的任务,需要找专业人士获取信息,并将信息拿回来分析综合完成任务。就像不同国家间语言不同,很难交流,但只要双方都有人会一种语言,如:英语,大家就可以开始谈生意了。事实上,英语差不多已成为世界通用语言。显然,在这样的架构下,既允许各个区域自由发挥——可根据自身的需要选择本地的 Ontology 和推导引擎,又保证了交流——使用统一的 Ontology,采用统一的通信模式。

当然,在这样的架构下予以实现还有其他好处:

◇极大地提高了 Agent 的处理速度。试想由自己的 Agent 为查找信息去一页页地分析别人的网页,就像在城市中找人时在大街上一个一个地问过来一样可笑。

◇这是符合面向 Agent 的软件开发趋势的。

◇为了使自己的信息更好地被理解,大家都有建设好自己的 Ontology 的动力。这将有力地推动 Ontology 的发展。

◇在各个区域允许各种 Web 和 Agent 技术的发展,有利于促进总体的技术水平提高。

◇最为重要的是,降低了最初实施的难度,能使其尽快行动起来。

有的时候,成功=正确的方向+合理的步骤。

参考文献

- Hendler J. Agents and the Semantic Web. IEEE INTELLIGENT SYSTEMS 1094-7167/01/\$ 10.00 (c)2001 IEEE, 2001, 16(2): 30~37
- McIlraith S A, Son T C, Zeng Honglei. Semantic Web Services. IEEE INTELLIGENT SYSTEMS 1094-7167/01/\$ 10.00 (c) 2001 IEEE, 2001, 16(2): 46~53
- Heflin J, Hendler J. A Portrait of the Semantic Web in Action. IEEE INTELLIGENT SYSTEMS 1094-7167/01/\$ 10.00 (c)2001 IEEE, 2001, 16(2): 54~59
- Fensel D. The Semantic Web And Its Languages. IEEE INTELLIGENT SYSTEMS 1094-7167/00/\$ 10.00 (c) 2000 IEEE, 2000, 15(6): 67~73
- Decker S, et al. The Semantic Web: The Roles of XML and RDF. IEEE INTERNET COMPUTING 1089-7801/00/\$ 10.00 (c) 2000 IEEE, 2000, 4(5): 63~74
- Wu Wongse V, et al. XML Declarative Description: A Language for the Semantic Web. IEEE INTELLIGENT SYSTEMS 1094-7167/01/\$ 10.00 (c)2001 IEEE, 2001, 16(3): 54~65
- Maedche A, Staab S. Ontology Learning for the Semantic Web. IEEE INTELLIGENT SYSTEMS 1094-7167/01/\$ 10.00 (c)2001 IEEE, 2001, 16(2): 72~79
- Dumbill E. Berners-Lee and the Semantic Web Vision, a keynote session at XML 2000 Tim Berners-Lee, Director of the World Wide Web Consortium, outlined his vision for the Semantic Web. [Dec. 6, 2000]
- Fensel D, et al. OIL: An Ontology Infrastructure for the Semantic Web. IEEE INTELLIGENT SYSTEMS 1094-7167/01/\$ 10.00 (c)2001 IEEE, 2001, 16(2): 38~45
- Berners-Lee T, Hendler J, Lassi O. The Semantic Web. Scientific American, May 2001
- Hendler J, McGuinness D L. The DARPA Agent Markup Language. IEEE Intelligent Systems, 2000, 16(6): 67~73
- Brickley D, Guha R V. Resource Description Framework (RDF) Schema Specification 1.0. W3C Candidate Recommendation 27 March 2000. <http://www.w3.org/TR/2000/CR-rdf-schema-20000327>
- De Giacomo G, et al. ConGolog, a concurrent programming language based on the situation calculus; Language and implementation. Submitted, 1999
- McDermott D, Burstein M, Smith D. Overcoming Ontology Mismatches in Transactions with Self-Describing Service Agents. In: Proceedings of SWWS' 01 The First Semantic Web Working Symposium Stanford University, California, USA July 30 - August 1, 2001
- Bray T, et al. Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation 6 October 2000. <http://www.w3.org/TR/2000/REC-xml-20001006>
- Malhotra A, Maloney M. XML Schema Requirements W3C Note 15 February 1999. <http://www.w3.org/TR/NOTE-xml-schema-req>
- Lassila O, Swick R. Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999. <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>
- Berners-Lee T. Semantic Web - XML 2000. <http://www.w3.org/2000/Talks/1206-xml2k-tbl/>
- Klein M. XML, RDF, and Relatives. IEEE INTELLIGENT SYSTEMS 1094-7167/01/\$ 10.00 (c)2001 IEEE, 2001, 16(2): 26~28
- Noy N F, et al. Creating Semantic Web Contents with Protégé-2000. IEEE INTELLIGENT SYSTEMS 1094-7167/01/\$ 10.00 (c)2001 IEEE, 2001, 16(2): 60~71
- Gordon M J C, Melham T F. Introduction to HOL: A theorem proving environment for higher order logic. Cambridge University Press, 1993
- Sutcliffe G, Suttner C. The TPTP Problem Library for Automated Theorem Proving. <http://www.cs.miami.edu/~tptp/>
- IBM Research's Business Rules for E-Commerce project team, IBM releases CommonRules 1.0: business rules for the Web. <http://www.research.ibm.com/rules/>
- Wooldridge M, Jennings N R. Intelligent Agents: Theory and Practice. Submitted to Knowledge Engineering Review, Oct. 1994. Revised Jan. 1995
- Eliens A, et al. Web Agent Support Program. <http://www.cs.vu.nl/~eliens/WWW6/papers/wasp/>
- Connolly D, et al. Annotated DAML + OIL Ontology Markup. W3C Note 18 Dec. 2001. <http://www.w3.org/TR/2001/NOTE-daml+oil-walkthru-20011218>
- Jennings N R, Wooldridge M. Agent-Oriented Software Engineering. Department of Electronic Engineering Queen Mary & Westfield College