

基于 Web Services 技术的 MAS 协作机制实现^①

The Realization of Web Services - Based MAS Cooperative Mechanism

刘慧敏 王万森 (首都师范大学信息工程学院 100037)

谭 娜 (河南大学计算机与信息工程学院 475001)

摘要:协作机制的研究是多 Agent 系统的基础。本文提出了基于 Web Services 技术的 MAS 协作机制的实现方案,该方案不仅完善和丰富了多 Agent 系统协作机制理论,也为开发实际的多 Agent 系统提供了有效的处理方法和实现技术。

关键词:Web Services Agent 多 Agent 系统 协作

1 引言

MAS 协作机制是 Agent 技术理论研究的核心问题之一。虽然针对不同的应用领域,已经提出多种 MAS 协作机制,但大部分只是建立了理论模型,在实际系统中的应用比较少。本文介绍了基于 Web Services 技术实现 MAS 协作机制的可行性、优势以及实现的基本方案。该方案不是为某一种协作机制提出的,它是实现协作机制的整体框架,针对不同的协作机制的特性,可进行不同的修改或扩充。

2 相关技术

本研究涉及的关键技术包括 SOAP、Web Services 及 MAS 协作机制。本部分介绍这些技术,并分析它们之间相互的联系。

2.1 SOAP

SOAP 以 XML 形式提供了一个简单、轻量级的用于在分散或分布环境中交换结构化和类型化信息的机制。SOAP 采用 HTTP 作为底层通信协议,XML 作为数据传送的格式。SOAP 本身并没有定义任何应用程序语义,它通过提供一个有标准组件的包模型和在模块中编码数据的机制,定义了一个简单的表示应用程序语义的机制^[3]。这使 SOAP 能够被应用于从消息传递到 RPC 的各种系统,使得我们完全能实现异构平台间的组件调用和通信。

2.2 Web Services

简单地说,Web Services 是通过标准的 Web 协议可编程访问的 WEB 组件。涉及的主要规范与协议包括:

XML — 用于交换结构化数据的通用标准,它构成了 Web Services 的基础;

组件发现 UDDI — 可通过 Web 访问的 Web 服务的注

册中心。其注册表提供了“一次注册、到处发布”的方式访问 Web Services 信息;

组件接口描述 WSDL — 基于 XML 的 Web 服务描述语言,用来定义 Web Services 并描述如何访问它们;

组件访问 SOAP — XML 格式的消息交换协议。

2.3 MAS 协作机制

Agent 是一个自主的程序,它能基于其对环境的理解,有能力控制自己的决策行为,以达到一个或多个目标。多 Agent 系统(MAS)是由多个问题解决者(Agent)组成的松散的网络,其中的 Agent 相互作用从而解决单个 Agent 由于能力或知识上的不足而无法解决的问题。对协作行为的研究是多 Agent 系统的基础。在多 Agent 系统中,Agent 群体协作行为的协调程度、系统的全局连贯性以及多 Agent 组织是否具有自适应性,是关系到整个系统性能和开放性的核心研究内容,同时也是衡量系统智能水平的重要指标^[1]。

目前,针对不同的应用,已提出多种协作机制,例如同联网、黑板系统、市场机制、多 Agent 规划、多 Agent 组织等。这种协作机制可通过协议的形式编程到 Agent 中,从而使 Agent 自动实现大型复杂问题的协作求解。虽然协作机制与具体的应用密切相关,但实现上的要求基本是相同的,如要求一种发现机制、Agent 间的交互与通信等。

3 基于 Web Services 的多 Agent 协作方法

3.1 Web Services 的优势

首先,通过运用 Web Services 所使用的 XML、SOAP、WSDL、UID 等标准规范与协议,我们可以建造能够“生存”在 Intranet 甚至 Internet 中的 Agent。而且在充分利用 Internet 提供的比较成熟的网络体系结构及通信协议的基

① 基金项目:北京市教委科技项目(2002kj098)

基础上,实现 Agent 间的协作更加简单、快速和方便。其次,如果用户频繁地从某个 Agent 中获取知识(信息),Web Services 会将这些知识保存到缓存中,这样当遇到相同的调用请求时,就可以直接返回结果,提高了运行效率。再者,Web Services 网络拥有许多 Agent,各 Agent 具有不同的领域知识,当一个 Agent 的知识与其他 Agent 的知识发生冲突时,Web Services 能够判断并挑选出最合适的知识发送给请求 Agent,这样方便地解决了 Agent 协作过程中的冲突问题^[4]。

3.2 MAS 与 Web Services

一方面,Agent 技术是一项很理想的技术,然而这项技术很少能走出实验室,能商业化和产业化。除了有部分难点有待解决以外,关键的问题在于 Agent 技术一直游离在当前主要的软件体系架构之外,从而得不到实际的应用。当今存在三种主要的软件体系架构:CORBA, J2EE, .NET,这三种技术有不同的组织(公司)所提出,使得相互间的通信十分困难。Web Service 的出现改变了这一状况,由于 Web Service 采用了统一的通信协议——SOAP 和标准的数据格式——XML,不管采用何种技术所构建的工程应用都能够为外部提供自身的服务,从而使得相互间的通信成为可能。所以,将 Agent 技术与 Web Service 结合是一种较合理的选择^[5]。

另外,多 Agent 系统(MAS)是由多个问题解决者(Agent)组成的松散的网络,其中的 Agent 相互作用从而解决单个 Agent 由于能力或知识上的不足而无法解决的问题。MAS 的主要特点在于,其中的每个 Agent 都不具备解决问题的足够的的能力或知识,这些 Agent 同时运行,不但所需的数据是分散的,而且没有全局控制系统。利用 Web Services 实现 Agent 的协作机制,可以充分利用 Web Services 的成熟技术。首先,当单个 Agent 无法解决问题时,它将会寻找多 Agent 系统中其他 Agent 进行协作,为了快速查找选择合适的合作伙伴,我们可以将一个 Agent 作为系统黄页,该黄页 Agent 记录它认识的所有 Agent 名字列表,另一种方法是利用 Web Services 的 UDDI 技术提供目录服务。其次,要实现协作,要求了解各个 Agent 的功能或服务,一个有效的方法是利用 WSDL。最后,为了利用其他 Agent 拥有的领域知识,关键需要与其他 Agent 间的交互,Agent 间需要某种机制进行通信,在 Web Services 环境下,我们就可以利用通过处理 SOAP 消息来达到这个目的。

另一方面,对于利用 Web Services 的应用程序,必须发生以下行为:发布服务描述、查询或查找服务描述以及根据服务描述调用服务等。当前的技术是依靠人的干预执行这些行为任务的。例如,UDDI 虽然解决了在 Internet 上如

何发布、查找、定位相应服务等问题,但还需要用户与服务提供者建立联系或者浏览在 UDDI 规范中涉及的众多文档;WSDL 文档把 Web Services 定义为—组处理消息的网络端点或端口,但仅仅是语法层次的描述。因此,现在的 Web Services 只是一种 Internet-based 组件,从本质上来说,Web Services 还是一种静态的远程调用(RPC),还不能自动发现、组成和执行服务,缺乏主动性。所以如何增加 Web Services 的智能性、主动性和协作性,是很值得探讨和很有意义的一个问题。正是基于这样的目的,我们将 Agent 的一些思想注入到 Web Services 中去,使得原先的静态远程调用的方式变成一种智能性、主动性和协作性的调用。

3.3 Agent 通信的消息结构

Agent 间协作的前提条件是各 Agent 应该理解相互间的通信,这就需要一种规范来统一 Agent 间的通信。国外许多专家提出了一些通信标准,其中以知识查询和操纵语言(Knowledge Query and Manipulation Language, KQML)规范较为成熟。当前应用 KQML 的多 Agent 系统存在的问题在于:每一个具体的系统有自己独有的 KQML,因此在不同的环境系统中,该语言具有不同的语义。目前大多数基于 Agent 的系统是利用一个本体服务器协助 Agent 在不同的环境中具有相同的语义。近来,研究者们开始关注在 Agent 的通信中利用 XML 来解决这一问题。XML 提供了描述结构化数据的格式,可以通过独立运行程序的方法来共享数据;同时,XML 又是一种用来自动描述信息的新的标准语言,它能使计算机通过 Internet 的强大功能把信息传递到人类的各种活动中去。在充分利用 Web Services 技术的基础上,我们运用了一种基于 SOAP 协议的 KQML 通信实现方法^[3]。在此方法中,用 XML 来表示 KQML 语言的执行原语(Performatives),并把这些执行原语放在 SOAP 协议的 Body 之中,最后通过 SOAP 协议在 Agents 之间进行通信。这种方法的消息结构如图 1 所示。

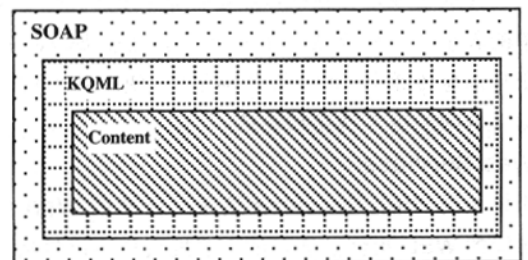


图 1 基于 SOAP 协议的消息结构

消息由三层组成:Soap 层、KQML 层和内容层。KQML 原语包含在 Soap 协议的 Body 中,由于消息语义的内容在 KQML 中规定并不详细,content 的内容可以用通信双方识别的任意语言书写。下面给出一个 KQML 执行原语的例子,

然后给出如何将它转换成 XML 表示并封装到 Soap 协议的 Body 中。

例如 Agent A 发送下面执行原语给 Agent B。

```
(ask-if
:sender A
:receiver B
:in-reply-to id0
:reply-with id1
:language prolog
:ontology foo
:content "bar(X,Y)")
```

Agent A 想知道 Agent B 是否认为内容 bar(X,Y) 正确。将该执行原语转换成 XML 表示并封装到 Soap 协议的 Body 中(此例中, Soap 消息是嵌入在 HTTP 请求中), 根据图 1 的消息结构, 上述原语的实现如图 2 所示。

4 结束语

采用基于 Web Services 的网络化软件体系架构技术, 可以更好地利用 Internet 这一分布、并行环境, 为 MAS 协作机制的实现提供了合理的解决方案。我们已经将此方案应用在专家系统平台的开发中, 形成一种基于 MAS 的专家系统开发平台^[2]。今后的工作重点是进一步完善和优化协作策略, 提高各 Agent 的交互的效率和性能等。

参考文献

- 1 史忠植, 智能主体及其应用, 科学出版社, 2001。
- 2 王万森, 人工智能原理及其应用, 电子工业出版社, 2000。
- 3 冯志勇、洪卫林, 基于 SOAP 协议的 KQML 语言通信实现, 计算机工程, 2003(4): 97-98。
- 4 John Fou. Web Services and Mobile Intelligent Agents. <http://www.webservicesarchitect.com/content/articles/fou02.asp>.
- 5 <http://www.ws-c.org/forum/cgi-bin/topic.cgi?forum=7&topic=6&start=0&show=>'。

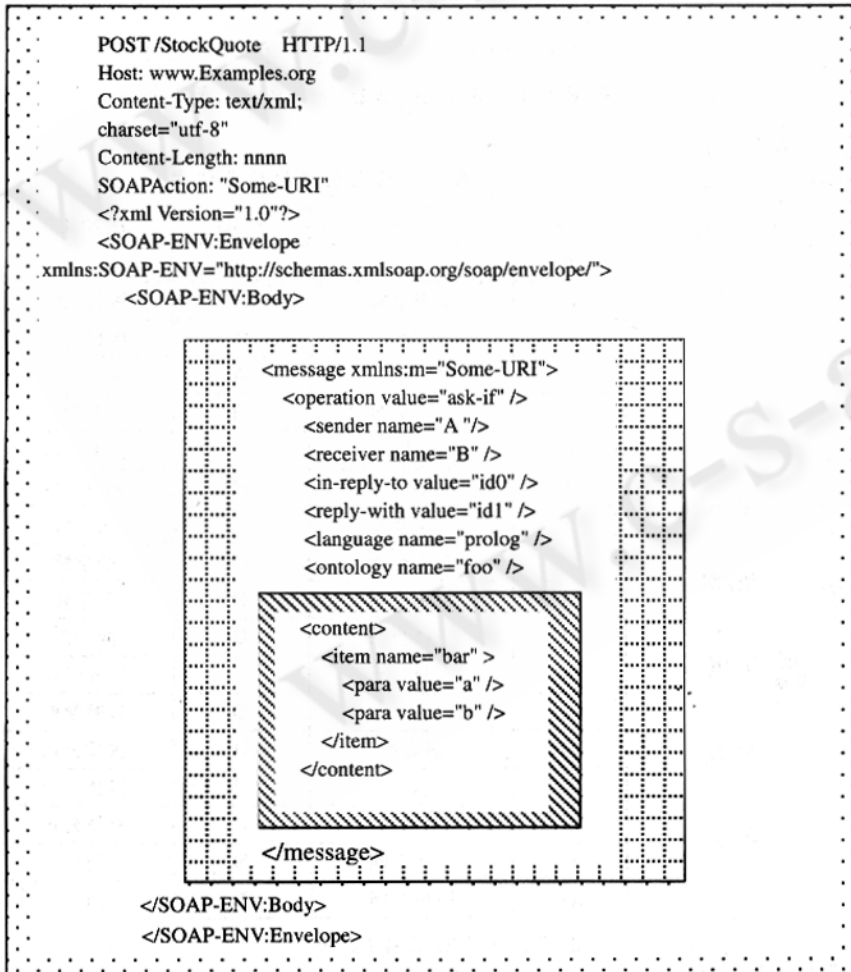


图 2 消息通信实例