

一个基于 XML 和 CORBA 的 Agent 通信框架

叶 斌

(湖北民族学院 信息工程学院, 湖北 恩施 445000)

摘 要: Agent 通信机制是多 Agent 系统中 Agent 间相互协作、完成任务的基础和关键。文中在 Agent 通信语言模式的基础上提出并设计了一个基于 XML 和 CORBA 技术的 Agent 通信框架, 着重对该系统的原理、体系结构、模块功能以及其优点作了分析和描述。最后, 结合 KQML 通信原语探讨了利用 XML“封装”KQML 消息的方法和作用。

关键词: Agent; 通信; 知识查询及操纵语言; XML; CORBA

中图分类号: TP393

文献标识码: A

文章编号: 1673-629X(2006)12-0032-03

An Agent Communication Framework Based on XML and CORBA Technique

YE Bin

(School of Information Engineering, Hubei Institute for Nationalities, Enshi 445000, China)

Abstract: Agent communication mechanism is the basis and the key element for agent coordination, cooperation, and task-performing in multi-agent system. Based on agent communication languages model, present an agent communication framework based on XML and CORBA technique, and analyze the principle, architecture, function and benefit of it. At the end, based on KQML communication primitive languages, also discuss the method and the effect of KQML was wrapped up in XML.

Key words: agent; communication; KQML; XML; CORBA

0 引 言

自主的智能 Agent 及多 Agent 系统(MAS)代表了一种新的软件分析与设计, 以及实现复杂软件系统的新思路和新方式, 很可能成为下一代软件开发的重要突破点^[1,2]。在多 Agent 分布式智能系统中, 由于问题本身的复杂性、分布性和内部相关性, 多 Agent 之间的相互协作与协商模型及其策略, 仍然是多 Agent 系统中要研究的一个关键问题。而实现 Agent 间相互协作、共同完成求解任务的基础是 Agent 之间的通信, Agent 之间必须使用相互能够理解的语言进行通信, 即 Agent 通信语言(ACL)。有文献归纳了 Agent 系统中常见的 Agent 间合作及通信模式可分为五种, 它们分别是无通信模式、消息传递模式、方案传递模式、黑板模式和 Agent 通信语言模式等^[3]。前四种模式对于问题空间的语义和 Agent 通信的语义, 都难以明确地描述; 它们都需要与 Agent 自身的求解逻辑融合成一体才能实现通信任务。这是它们的共同缺点。而 Agent 通信语言模式使用“高级”的通信语言 ACL, 它能较好地描述 Agent 通信的语义, 可满足 Agent 间通信的基本

要求, 具有灵活性、通用性, 支持知识共享和 Agent 间的协作。目前, ACL 主要有两种: 一种是 1993 年由美国 DARPA(Defense Advanced Research Project Agency)的 KSE(Knowledge-Sharing Effort)研究小组提出的知识查询及操纵语言 KQML(Knowledge Query and Manipulation Language); 另一种则是 1997 年由欧洲的智能代理基础组织 FIPA(Foundation for Intelligent Physical Agents)提出的 FIPA-ACL。虽然还有一些其他语言出现, 但这两种语言版本基本上已成为 Agent 间通信的语言标准, 代表着 Agent 通信领域的主流方向(尽管它们都还存在一些缺点和局限性, 还需要完善), 关于二者的定义和比较在此不作赘述。文中在 Agent 通信语言模式的基础上, 结合 XML 和 CORBA 的技术特性, 构建了一个基于 XML 和 CORBA 的 Agent 通信模型, 下面介绍其原理和体系结构。

1 基于 XML 和 CORBA 的 Agent 通信框架

构建基于 XML 和 CORBA 的 Agent 通信框架, 其主要思路是: 通过采用先进的 CORBA 规范来实现 KQML; 利用 XML 技术实现 ACL 与 WWW 的集成, 将 ACL 消息和本体知识嵌入到 XML 中去, 从而达到解决系统的异构性和互操作性问题。

1.1 系统原理

XML 是 W3C 制定的用于描述数据文档中关于数据

收稿日期: 2006-06-01

基金项目: 湖北省教育厅优秀中青年资助项目(Q200629001)

作者简介: 叶 斌(1962-), 男(土家族), 湖北建始人, 副教授, 研究方向为计算机软件、数据库系统。

组织结构的标记语言。XML 作为一种互联网上交换数据的标准,其目标是让各种结构的文档都作为统一的网络文件的一部分在网上传输,以实现在不同的系统间进行数据交换。XML 具有可扩展性、平台无关性、结构化语义、数据无缝集成以及面向数据而独立于应用等特点。利用这些特性可以充分满足 Internet 和分布式异构环境的要求,将成为下一代网络数据传输的主要载体,有利于多 Agent 系统中异构性问题的解决,非常适用于异质 Agent 之间通信语言的描述。

CORBA(Common Object Request Broker Architecture, 公共对象请求代理体系结构)是由 OMG 组织制定的一种标准的面向对象应用程序体系规范。CORBA 独立于网络协议,独立于编程语言,独立于硬件平台。CORBA 提供了一种 ORB(Object Request Broker)机制,通过这种机制,对象间可以透明地进行通讯和访问。通过 ORB 应用框架,使得异地机器、异质平台、异种语言开发的对象之间可连接起来。CORBA 定义了一组标准的接口规范 IDL 来提供统一的接口描述语言,每个对象都将其内部操作细节封装起来,同时又向外界提供了精确定义的接口,使不同的应用可以共享由 CORBA 规范构造出来的软件构件。如果基于 CORBA 规范实现 KQML,则有关 CORBA 的突出特性便可自然应用到 Agent 通信中去。

ACL 是一种高级语言,它的原语和结构专用于支持 MAS 中 Agent 的合作、协商和信息传输。用 ACL 实现 Agent 通信,需要具备下述前提条件:信息传输媒介;通信的起因;Agent 寻址机制;内容定义语言和 Agent 通信语言。在 MAS 中 Agent 通信一般可分为三个层次,自下而上分别是:传输层、通信语言层和会话层。通信过程中的消息可划分为两个层面,即通信原语层和通信内容层。考虑将 XML 应用于 ACL,可用 XML 来“封装”通信原语消息,同时也可用 XML 来表示通信的内容。图 1 所示即为 Agent 利用 ACL 通信的层次结构^[3]。

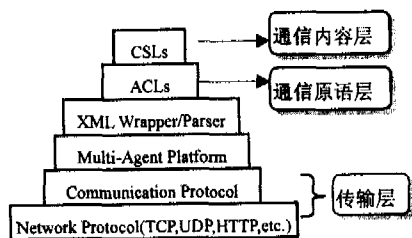


图 1 Agent 通信层次结构

1.2 系统体系结构

基于上述技术背景和对 Agent 通信原理的分析,文中所构建的基于 XML 和 CORBA 的 Agent 通信模型是建立在中间件 CORBA 的基础之上,在传输层利用 CORBA 提供传输协议支持,以便 MAS 中各个 Agent 能顺畅地穿越各异构的系统和协议进行通信,为了达到这个目的,CORBA 使用 IDL 语言定义的接口,使其在接口语法上达成一致。Agent 通信语言采用 KQML,它是 XML 之上的通信层的语言,可以描述关于 XML 所标识信息的请求动作

类型,它独立于任何实现技术,既可用 C、C++ 或 Java 在 TCP/IP 上实现,也可用 CORBA 规范来实现。另外,用 XML 来“封装”KQML 通信原语消息,同时也用 XML 来表示通信的内容。系统功能体系结构如图 2 所示。

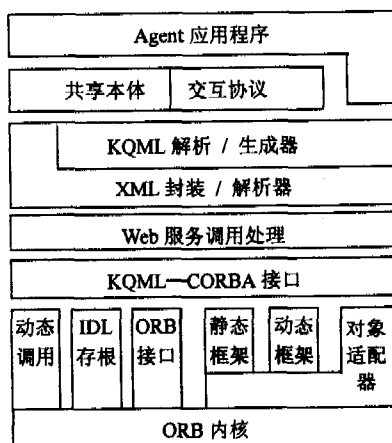


图 2 系统功能体系结构

1.3 系统功能描述

在图 2 中,ORB 作为软总线,提供了分布对象通信的基础设施;CORBA 主要用于将底层的、繁杂的网络通信任务封装起来,为 Agent 间的通信提供一个平台。CORBA 服务对象可由对象适配器发现并获得,服务对象各模块分别实现相应 IDL 定义的消息服务;KQML-CORBA 接口识别 KQML 执行词,将其参数转换成 CORBA 可以处理的类型,并根据 CORBA 对象的类型调用与执行词相匹配的方法;Web 服务调用处理模块负责截取 KQML 消息,识别并处理其中服务请求的消息;XML 封装/解析器主要用于对 KQML 语言串的消息进行结构化表示,即它负责将 KQML 语言串生成 XML 文档,或者接收 XML 文档信息,并从中分离出 ACL 消息;KQML 解析/生成器用于将请求生成 KQML 语言串,或者接收以 XML 形式表达的 KQML 语言串,结合选用共享本体将它们转化成相应的结果并生成应答。通过 XML 解析器和 KQML 解析器来实现编码和解码。

2 系统中使用的通信语言

本系统使用 KQML 实现 MAS 中 Agent 间的通信语言。系统提供对 KQML 消息的封装功能,即用 XML 将 KQML 语言串封装成 XML 文档作为要传输的数据。由于目前已有许多成熟的 XML 文件解析器,因而可以利用现有的 XML 文件解析器来封装或解析 KQML 消息,从而免去了编写解析器的工作。

2.1 KQML 的保留通信原语

KQML 是一个高层的、面向消息的 Agent 通信语言和通信协议。它独立于下层的传输机制和 Agent 内容语言以及 Agent 内容定义的本体。KQML 中每个 Agent 都拥有和管理一个表示其精神状态模型的知识库(KB, Knowledge Base)。KQML 预定义了一套丰富的、可扩展

的原语,它们可完成 Agent 交互中通信要求的基本操作。KQML 的保留通信原语见表 1。

表 1 KQML 的保留通信原语

功能分类	通信原语名称
会话类	ask - all, ask - one, ask - if, delete - all, delete - one, undelete, tell, untell, deny, insert, uninsert, stream - all, achieve, unachieved, advertise, unadvertised, subscribe
会话干预与机制原理	sorry, error, standby, ready, next, rest, discard
网络服务类	register, forward, broadcast, transport - address, broker - all, broker - one, recruit - all, recruit - one, recommend - all, recommend - one

部分功能说明:

* ask - all: sender 希望知道 receiver 中: content 为真的所有实例;

* delete - all: sender 请求从 receiver 的知识库中删除所有满足条件的记录;

* tell: sender 向 receiver 表明: content 在 sender 中为真;

* insert: sender 要求 receiver 向其知识库中写入: content; 等等。

2.2 KQML 的消息语法

KQML 语言的一般格式可参见文献[4], KQML 的消息语法, 采用类似于 LISP 的语法描述, 开头以一个通信原语(performative)作为第一行, 后跟参数名称(parameter name)和参数值(parameter value)。下面是一般的 KQML 消息语法描述。

(performative

```
:sender <word> // 消息发送方
:receiver <word> // 消息接收方
:reply_with <word> // 消息的 ID
:in_reply_to <word> // 所回答消息的 ID
:language <word> // 内容语言
:ontology <word> // 本体名称
:content <expression> // 消息内容
)
```

2.3 用 XML 封装 KQML 消息

由于 KQML 采用一种类似于 LISP 的语法来描述 ACL 消息, 所以 KQML 消息的内容本身可以用 XML 来编码, 用 XML 编码后更容易编写解析器, 而且用 XML 编码的消息还有利于在不同的操作系统平台上传输和解析。以下是将 KQML 消息直接封装成 XML 文档的一个例子[5]。

• Agent Peter 要发送的 KQML 消息为:

Advertise

```
:sender Peter :reply_with q1 :language KQML :ontology Kn
```

Kn

```
:content (Evaluate :language KQML)
```

```
:ontology P1 :reply_with q1
```

```
:content (val(torque motor 1)(sim-time 5))
```

• 用 XML 文件解析器将要发送的 KQML 消息生成如下的 XML 文档:

```
<! - advertise.xml - - >
```

```
<? xml version="1.0" encoding="GB2312">
```

```
<KQML>
```

```
Advertise :sender Peter :reply_with q1 :language
KQML :ontology Kn
```

```
:content (Evaluate :language KQML) :ontology P1 :reply
-with q1
```

```
:content (val(torque motor 1)(sim-time 5))
```

```
</KQML>
```

3 结 论

ACL 是多 Agent 系统中 Agent 间相互协作、共同完成求解任务的基础和关键。文中提出并设计的基于 XML 和 CORBA 技术的 Agent 通信框架, 通过采用先进的 CORBA 规范来实现 KQML、利用 XML 来描述 KQML 消息, 给 KQML 消息的编码、解析和可扩展性带来了便利, 有利于解决 MAS 的平台异构性和互操作性问题, 为 Agent 通信机制提供了一个有效的技术实现方案。

参考文献:

- [1] Jennings N R, Sycara K, Wooldridge M. A roadmap of agent research and development[C]// Autonomous Agents and Multi-Agent System. Boston: Kluwer Academic Publishers, 1998: 275-306.
- [2] 石 慧, 徐从富. Agent 通信语言 KQML 的实现及应用[J]. 计算机工程与应用, 2005, 41(13): 94-97.
- [3] 魏晓斌, 周盛宗. Agent 通信机制探讨[J]. 计算机工程与应用, 2002, 38(5): 66-70.
- [4] Finin T, Labrou Y, Mayfield J. KQML as an agent communication language[EB/OL]. 1996. Draft; University of Maryland Baltimore County, <http://www.cs.umbc.edu>.
- [5] 邱 建. Internet 上多 Agent 之间通信框架的研究[D]. 武汉: 武汉理工大学, 2005.

(上接第 31 页)

- [2] Singh I, Stearns B, Johnson M, et al. Designing Enterprise Application with J2EE platform[M]. 2nd ed. London: Addison-Wesley, 2002.
- [3] Sun Microsystems, Inc. J2EE Specification[EB/OL]. 2005-06-20. <http://java.sun.com/J2EE/>.

- [4] Sun Microsystems, Inc. JMX Specification[EB/OL]. 2002. <http://java.sun.com/products/JavaManagement/>.
- [5] AdventNet Inc. AdventNet Agent ToolKit[EB/OL]. 2005. <http://www.adventnet.com/products/javaagent/help/snmp-agent/javadocs/index.html>.