

文章编号:1001-9081(2005)03-0539-04

基于 KQML 面向虚拟企业合作伙伴选择的多 Agent 系统通信模型

谢家友, 曾建潮

(太原科技大学 系统仿真与计算机应用研究所, 山西 太原 030024)

(xiejiayou01@sina.com)

摘要:给出了一个面向虚拟企业合作伙伴选择的多 Agent 系统基于 KQML 的通信模型。介绍了各 Agent 间通信时用到的 KQML 保留原语, 并对 KQML 原语进行了适当的扩充以满足系统通信的要求。最后给出了在通信中使用较多的与数据操作有关的 KQML 消息到相关的数据操作 SQL 语句间的映射关系, 较好地实现了 KQML 消息与 SQL 语句的衔接。

关键词:知识查询操纵语言; 多 Agent 系统; 原语; 本体论; SQL

中图分类号: TP393.01 **文献标识码:** A

Multi-agent system communication model for KQML-based virtual enterprise oriented partner selection

XIE Jia-you, ZENG Jian-chao

(Division of System Simulation & Computer Application, Taiyuan University of Science & Technology, Taiyuan Shanxi 030024, China)

Abstract: KQML has been accepted as a standard of ACL (Agent Communication Language), and it offers knowledge-based communication between agents. This paper presented a multi-agent system communication model of virtual enterprise oriented partner selection based on KQML, including reserved performatives, extended performatives for system communication and necessary maps between some KQML messages and SQL sentences.

Key words: KQML (Knowledge Query and Manipulation Language); MAS; performative; ontology; SQL

0 引言

KQML (Knowledge Query and Manipulation Language) 是基于 Speech Act 理论的知识交换语言。由 KSE (Knowledge Sharing Effort) 机构开发。它是自主的异步主体间共享知识和实现协作问题求解的通信语言。KQML 既是一种消息格式, 也是支持实时主体间知识共享的消息处理协议, 其目的在于在基于知识的异质系统间实现互操作和集成^[1]。它是软件 Agent 通信语言事实上的标准。KQML 预定义了一套丰富的原语 (performative), 能够满足 Agent 间大多数的通信要求; 同时 KQML 还是可扩展的, 可以针对特定的应用领域定义特定的 KQML 原语来满足 Agent 间特殊的通信要求。目前, 有非常多的 MAS 系统均采用 KQML 作为通信语言。

一条典型的 KQML 消息如下:

```
(performative
: sender <word>           //消息的发送者
: receiver <word>          //消息的接收者
: in-reply-to <word>
      //应答标记, 与前一条消息的: reply-with 值相同
: reply-with <word>        //在回复本条消息时要求的应答标记
: language <word>          //: content 内容的表示语言名
: ontology <word>          //: content 参数中使用的术语定义的集合
: content <expression>)     //执行原语要表达的态度信息
```

1 虚拟企业合作伙伴选择的系统模型

图1就是我们所要介绍的一个关于虚拟企业合作伙伴选

择的 MAS 实验系统, 它站在盟主企业的角度, 将在合作伙伴选择过程中盟主企业所要具备的功能分解成多个 Agent 来分别实现。该系统由 5 个 Agent 组成: 用户 Agent (UserAgent)、信息 Agent (InfoAgent)、数据库 Agent (DBAgent)、评价 Agent (EstiAgent) 和协商 Agent (ConsAgent)。我们使用 KQML 消息来实现各个 Agent 间的通信。

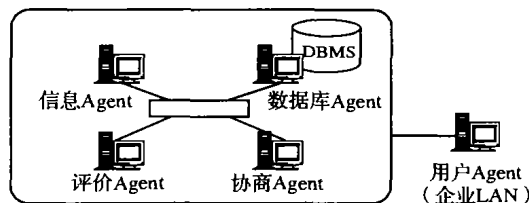


图1 合作伙伴选择的 MAS 系统

盟主企业通过 UserAgent 将一些原始数据和对合作伙伴的要求写入 DBAgent 的数据库中, InfoAgent 从 DBAgent 中提取相关数据发布到企业网站开始进行数据的收集工作, 并将收集到的企业数据资料写入 DBAgent 的数据库中。当收集到一定的数据后, EstiAgent 从 DBAgent 中提取相关的数据进行评价, 对评价后筛选出的候选企业由 ConsAgent 与相关企业进行协商。通过反复的评价与协商为盟主企业找到合适的合作伙伴, 构建虚拟企业。本文给出了该 MAS 建立在 KQML 语言基础上的通信模型。

2 系统中用到的通信原语

2.1 模型中用到的 KQML 保留原语

首先介绍一下通信模型中用到的 KQML 保留原语:

收稿日期: 2004-08-19; 修订日期: 2004-11-04

作者简介: 谢家友 (1975-), 男, 湖北宜昌人, 硕士研究生, 主要研究方向: 人工智能; 曾建潮 (1963-), 男, 陕西大荔人, 博士生导师, 主要研究方向: 智能计算。

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

2) delete-all, sender 请求从 receiver 的 KB 中删除所有满足条件的记录;

3) delete-one, sender 请求从 receiver 的 KB 中删除一条满足条件的记录;

4) insert, sender 要求 receiver 向其 KB 中写入:content;

5) tell, sender 向 receiver 表明:content 在 sender 中为真;

6) forward, 用于从 sender 向 receiver 发送来自 from 目的地为 to 的请求。

2.2 自定义原语

根据模型的需要,我们对 KQML 进行了必要的扩充,我们对扩充的原语参数均使用 KQML 的保留参数: sender, receiver, in-reply-to, reply-with, language, ontology 和:content。因此下面对原语的定义中我们省略了对原语参数的定义,只对自定义原语进行语法描述和语义分析。在描述 KQML 原语的语义时我们使用表达式的集合,这些表达式描述了信息交换过程中主体的状态或行为,其意义如下^[1]:

BEL 形式为 BEL(A,P),意指主体 A 相信 P 为真(或可以被证实),P 是主体 A 的语言中的一个表达式。

KNOW 形式为 KNOW(A,S),表示主体 A 知道 S 描述的知识,这里 S 是一个状态描述(以下的 S 同此义)。

WANT 形式为 WANT(A,S),表示主体 A 期望 S 描述的认知状态(或行为)在将来发生。

INT 形式为 INT(A,S),表示主体 A 有从事 S 的意图,并且承诺一个行为过程在将来达到 S。

PROC(A,M) 指主体 A 处理 KQML 消息 M 的行为,但 PROC(A,M)并不保证对消息的恰当处理。

SENDMSG(A,B,M) 指主体 A 把 KQML 消息 M 传给主体 B 的行为。

2.3 对扩充原语的定义及语法、语义描述

2.3.1 order(A,B,X)

语法描述:

```
(order
: sender < word >
: receiver < word >
: reply-with < word >
: language < word >
: ontology < word >
: content < expression >)
```

语义描述:

1) 对原语的自然语言表述:A 希望 B 完成任务 X。

2) 对原语的形式化表示:WANT(A,INT(B,X))。

3) Pre(A):WANT(A,INT(B,X))。

Pre(B):无。

4) Post(A):WANT(A,KNOW(B,WANT(A,X)))。

Post(B):KNOW(B,WANT(A,X))。

5) Completion:INT(B,X)。

2.3.2 request-begin(A,B,X)

语法描述:

```
(request-begin
: sender < word >
: receiver < word >
: reply-with < word >
```

```
: language < word >
```

```
: ontology < word >
```

```
: content < expression >)
```

语义描述:

1) 对原语的自然语言表述:A 想 B 知道 A 有从事 X 操作的意图。

2) 对原语的形式化表示:WANT(A,KNOW(B,INT(A,X)))。

3) Pre(A):WANT(A,KNOW(B,INT(A,X))) ^ KNOW(A,¬KNOW(B,INT(A,X)))。

Pre(B):¬KNOW(B,INT(A,X))。

4) Post(A):INT(A,X)。

Post(B):KNOW(B,INT(A,X))。

5) Completion:KNOW(B,INT(A,X))。

2.3.3 request-end

语法描述:

```
(request-end
: sender < word >
: receiver < word >
: in-reply-to < word >
: language < word >
: ontology < word >
: content < expression >)
```

语义描述:

1) 对原语的自然语言表述:A 希望 B 结束 X 操作。

2) 对原语的形式化表示:WANT(A,¬INT(B,X))。

3) Pre(A):WANT(A,¬INT(B,X)) ^ KNOW(A,INT(B,X))。

Pre(B):INT(B,X)。

4) Post(A):WANT(A,KNOW(B,WANT(A,¬INT(B,X))))。

Post(B):KNOW(B,WANT(A,¬INT(B,X)))。

5) Completion:¬INT(B,X)。

2.3.4 new-advertise

该原语是在 KQML 保留原语 advertise 的参数中加入了一个:in-reply-to 关键字而形成的一条自定义原语。它的语义与 advertise 完全相同,改造这条原语的目的是使其能够响应其他 KQML 消息,而 advertise 不具备响应的功能,不能满足本系统的要求。

3 Agent 间的通信协议

为了便于描述各 Agent 间的通信协议,我们对问题的描述进行以下约定:

1) 首先定义一个 ontology:

```
term = { tableName1, tableName2, ..., tableNamem,
task1, task 2, ..., taski}
```

其中:tableNameX 表示数据表名(X=1,2,...,m);

taskZ 表示任务名(Z=1,2,...,i)。

2) KQML 参数:language 遵循 KIF 格式^[3]。

3) 在以下书写 KQML 消息时,我们不再采用引言中的标准形式,而是采用下面的省略形式^[2]:

```
(performative ... : content < expression >)
```

下面以表格的形式给出各 Agent 间的基于 KQML 的通信协议。

表 1 UserAgent 与 DBAgent 的通信协议

通信目的	UserAgent	DBAgent
UserAgent 向 DBAgent 发出写入、修改、查询或删除请求	(request-begin ... : content(tableNameX taskZ))	(new-advertise ... : content(insert, ask-all, ask-one, insert, delete-one 或 delete-all ... : content()))
UserAgent 向 DBAgent 中写入或修改数据	(insert ... : content (tableNameX (?field1 v1) (?field2 v2) ... (?fieldn vn)))	写入数据
UserAgent 从 DBAgent 中查询数据	(ask-all 或 ask-one ... : content (tableNameX (condition)))	(tell ... : content(result))
UserAgent 从 DBAgent 中删除数据	(delete-one 或 delete-all ... : content(tableNameX (condition)))	执行删除数据操作
UserAgent 向 DBAgent 请求结束写入、修改、查询或删除操作	(request-end ... : content(taskZ))	执行结束操作
UserAgent 通过 DBAgent 向 InfoAgent, EstiAgent 和 ConsAgent 分别发出信息发布、评价和协商命令	(forward : form : to ... : content(order ... : content(taskZ)))	执行转发命令

表 2 InfoAgent 与 DBAgent 的通信协议

通信目的	InfoAgent	DBAgent
InfoAgent 向 DBAgent 发出读或写请求	(request-begin ... : content(tableNameX taskZ))	(new-advertise ... : content (ask-all 或 insert ... : content()))
InfoAgent 从 DBAgent 读取要发布的数据	(ask-all ... : content(tableNameX (condition)))	(tell ... : content(result))
InfoAgent 向 DBAgent 的 KB 中写入收集到的信息	(insert ... : content (tableNameX (?field1 v1) (?field2 v2) ... (?fieldn vn)))	执行写入数据操作
InfoAgent 向 DBAgent 发出结束读或写操作请求	(request-end ... : content(taskZ))	执行结束操作

表 3 EstiAgent 与 DBAgent 的通信协议

通信目的	EstiAgent	DBAgent
EstiAgent 向 DBAgent 发出读或写请求	(request-begin ... : content(tableNameX, taskZ))	(new-advertise ... : content (ask-all, ask-one 或 insert ... : content()))
EstiAgent 从 DBAgent 读取评估所需的数据	(ask-all 或 ask-one ... : content())	(tell ... : content())
EstiAgent 将评估结果写入 DBAgent	(insert ... : content())	(new-advertise ... : content(insert : content()))
EstiAgent 向 DBAgent 发出结束读或写操作请求	(request-end ... : content(taskZ))	执行结束操作
EstiAgent 向 DBAgent 发出启动 ConsAgent 对评价后的结果进行协商的建议	(forward : from : to ... : content(request-begin ... : content(taskZ)))	执行转发命令操作

表 4 ConsAgent 与 DBAgent 的通信协议

通信目的	ConsAgent	DBAgent
DBAgent 命令 ConsAgent 对评价过的数据开始协商		(order ... : content())
ConsAgent 向 DBAgent 发读或写请求	(request-begin ... : content(tableNameX, taskZ))	(new-advertise ... : content(ask-one, ask-all 或 insert ... : content()))
ConsAgent 从 DBAgent 读取数据	(ask-one 或 ask-all ... : content())	(tell ... : content())
ConsAgent 向 DBAgent 写数据	(insert ... : content())	执行写入数据操作
ConsAgent 向 DBAgent 发出结束读或写操作请求	(request-end ... : content(taskZ))	执行结束操作

4 KQML 消息到 SQL 语句的映射

在本系统的通信过程中,Agent 间的许多通信是为了实现

数据的读写,因此,我们对模型中 KQML 消息与 SQL 语句间的映射作一些说明:

1) (insert ... : content (tableNameX (? field1 v1) (?

field2 v2) ... (? fieldn vn))

该 KQML 消息表示 sender 希望向 receiver 的 KB 中插入或修改一条记录,如果 insert 消息中 tableNameX 的关键字的值在数据表 tableNameX 中不存在,则将 insert 消息转换成 SQL 的 INSERT 语句:

```
INSERT
INTO TableNameX( field1, field2, ..., fieldn)
VALUES ( v1, v2, ..., vn)
```

否则转换 SQL 的 UPDATE 语句:

```
UPDATE tableNameX
SET field1 = v1, ..., fieldn = vn
WHERE fieldN = keyValue
```

2) (ask-one 或 ask-all ... : content (tableNameX (condition)))

该 KQML 消息表示 sender 希望从 receiver 的 KB 中读取一条或一组记录,可以将其转换成相应的 SQL 的 SELECT 语句:

```
SELECT field1, ..., fieldn
FROM tableNameX
WHERE condition
```

3) (delete-one 或 delete-all ... : content (tableNameX (condition)))

该 KQML 消息表示 sender 希望 receiver 从其 KB 中删除一条或一组记录,可以将其转换成相应的 SQL 的 DELETE 语句:

```
DELETE
FROM tableNameX
WHERE condition
```

通过上述映射,可以较好地实现从 KQML 消息到 SQL 语

句的转换。

5 结语

KQML 语言是构建 MAS 系统的流行语言。它是一门严格的、灵活的知识交换语言。它的保留原语要求我们在使用时必须严格按照 KQML 规范定义,它的可扩展性增加了灵活性,可以更好地满足我们特定的需求。本文利用 KQML 语言给出的这个关于虚拟企业合作伙伴选择的 MAS 的通信框架,可以较好地实现本系统中各 Agent 间的通信,是对基于 KQML 语言的 MAS 通信的一次有益的尝试。

参考文献:

- [1] 张维明,姚莉. 智能协作信息技术[M]. 北京:电子工业出版社, 2001. 117-138.
- [2] LABROU Y, FININ T. A Proposal for a New KQML Specification [EB/OL]. <http://www.cs.umbc.edu/html/>, 2003-08.
- [3] GENESERETH MR. Knowledge Interchange Format [EB/OL]. <http://logic.stanford.edu/kif/specification.html>, 2003-08.
- [4] Cogoi L, MEA VD. Seamless Access to Databases through KQML in an Agent-enriched Web [EB/OL]. <http://citeseer.nj.nec.com>, 2003-11.
- [5] 石纯一,张伟. 多 Agent 系统引论[M]. 北京:电子工业出版社, 2003. 115-133.
- [6] FININ T, WEBER J. DRAFT Specification of the KQML Agent-Communication Language [EB/OL]. <http://www.cs.umbc.edu/html/>, 2003-08.

(上接第 532 页)

同,在速度较小时,时间间隔由固定计时器触发(此处为13s),因而发送字节数变化不大,随着速度增加,时间间隔逐渐缩短,且节点位置变化及更新加快,字节数量迅速增加。但考虑到它们引起的 IP 层字节数增长较慢以及位置服务性能的提高,网络层字节开销的代价是值得的。

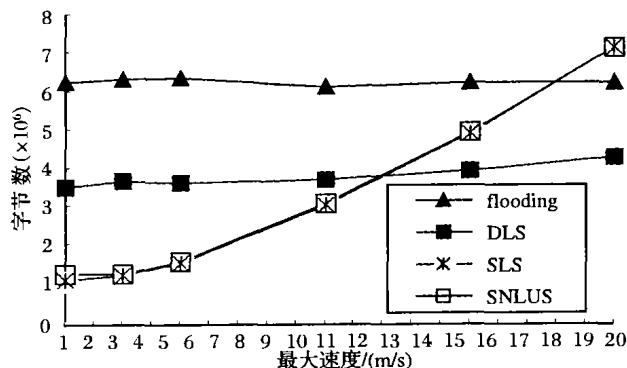


图3 最大速度—字节数开销关系

总之,与其他位置服务协议(Flooding, DLS, RLS)相比,SNLUS 协议提供了较好的性能(见图1)和较少的开销(见图2、图3)。和具有较高的性能和较少的开销的 SLS 算法^[4]相比,在没有增加开销的基础上大大提高了位置服务性能。

4 结语

在本文中,首先简单描述了5种位置服务,分析了各种位置服务的性能。相对来说,先应式的 SLS 具有较好的网络适应

性、较简单的算法、较精确的定位精度和较少的数据传输量。

为提高 ad hoc 网络位置服务的性能,我们在 SLS 协议的基础上,提出了一种简单新位置更新服务(SNLUS)。通过向邻居节点发送更新过的位置信息,逐渐使整个 Ad Hoc 网络中的每个节点都将全网节点的位置信息存储起来。它提供了一种适用于需要较精确的邻近主机位置信息的位置辅助型路由协议的位置服务,并可以在它的基础上,进行新的位置辅助型路由协议的研究。

参考文献:

- [1] KO YB, VAIDYA NH. Location-aided routing (LAR) in mobile ad hoc networks [A]. The ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM) [C]. Dallas, 1998.
- [2] BASAGNI S, CHLAMTAC I, SYROTIUK VI, et al. A distance routing effect algorithm for mobility (DREAM) [A]. The ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM) [C]. Dallas, 1998.
- [3] TSENG Y-C, WU S-L, LIAO W-H, et al. Location awareness in ad hoc wireless mobile networks [J]. Computer, 2001, 34(6): 46-52.
- [4] CAMP T, BOLENG J, WILCOX L. Location information services in mobile ad hoc networks [A]. Proceedings of the IEEE International Conference on Communications (ICC) [C], 2001.
- [5] LI JY, JANNOTTI J, De COUTO DSJ, et al. A scalable location service for geographic ad hoc routing [A]. Proceedings of the ACM/IEEE MOBICOM [C], 2000. 120-130.