

基于 Agent 协商机制的分布式资源库实现

上官晓清¹，赵文耘¹，彭鑫¹，肖君²

¹复旦大学计算机科学与工程系软件工程实验室，²上海远程教育集团

摘要：随着分布式资源库的广泛使用，如何在多个资源库中选出所需的资源以及不同资源库间怎样通过协商对资源交易达成共识，是分布式资源库下实现资源交易自动化的关键问题。本文提出一种基于 Agent 协商机制的分布式资源库解决方法。首先建立基于 Agent 架构的分布式资源库，然后将一种基于策略的 Agent 协商模型应用到本系统中，对于构造协商模型所需的协商信息，本文先通过将本体与刻面相结合的查询方法得到案例集，再由知识库对案例集进行案例推导得到所需的协商信息。

关键词：本体、刻面、案例推导、Agent、协商模型

项目资助：国家 863 计划(2004AA1Z2330, 2004AA112070, 2004AA113030)，国家自然科学基金(60473061)，上海市科委科研攻关项目(04DZ15022)。

1. 引言

分布式资源库作为解决信息资源迅猛增长的有效途径已经被广泛使用，许多基础应用平台，例如分布式软件构件库、分布式教育教学资源库等都是这种架构的成功应用。在有效解决了资源信息的存储这个问题的同时，如何在不同资源库中选择所需资源成为了一个重要的研究方向。对于分布式系统下的每一个资源库，都是相对独立的利益实体和决策实体。当一个资源在两个站点间进行交易时，双方都希望将自己的利益最大化。如何通过协商得到一个双方都可以接受的交易方案是解决这一矛盾的关键。

Agent 的代理性能很好的解决分布式系统下对资源站点的描述。多 Agent 之间的协商能力使得基于协商的跨站点资源请求成为可能，每个 Agent 可以代表所在站点的利益和资源管理策略参与到协商过程中，从而可以在资源拥有站点和资源索取站点间寻找到最大的共同利益。在前一阶段，我们参加了上海远程教育集团承建的 863 软件重大专项课题，这是一个多媒体网络教学资源管理和应用平台软件研究课题。在该课题的研究过程中，我们将不同资源子站点建立在多 Agent 架构系统之上，站点间对资源的请求通过移动 Agent 来完成，对教育资源实现了基于本体与刻面描述与检索。在资源交易过程中，我们首先通过子站点所拥有的知识库得到对所需资源的协商信息，然后由移动 Agent 根据这些协商信息构建自己的协商模型，再使用协商模型与能提供其所需资源的站点进行交易。

本文接下来在第 2 节中介绍基于多 Agent 系统的分布式资源库的实现和 Agent 在协商过程中使用的协商模型；在第 3 节中描述如何通过本体和刻面获取协商案例集，再结合知识库通过案例推导取得协商模型所需的协商信息；第 4 节通过一个实例说明如何通过基于 Agent 的协商机制来完成资源的自动交易；最后做出总结和对系统进一步完善的展望。

2. 基于 Agent 协商的分布式资源库

分布式资源库下的资源请求，即从大量的、异构的资源站点中获得所需资源的过程。

Agent 是指具有反应性、预动性、社会行为能力的软件实体[1]，多 Agent 系统下每一个 Agent 为了完成自己的任务都需要同其它 Agent 进行交互协作，本文中所讨论的是用于在一个分布式资源库中可以在不同站点对所需资源进行查找选购的多 Agent 系统，采用了由意大利 TILAB 设计的 JADE(Java Agent DEvelopment Framework)系统[2]，这是一个完全由 Java 语言实现的 Agent 系统，其 Agent 的构建完全满足 FIPA(The Foundation for Intelligent Physical Agents，这是一个制定软件 Agent 交互标准的一个组织)规范，支持移动 Agent。Agent 之间通过 ACL (Agent 通信语言) 进行交互，这种特性为在资源库之间进行协商提供了基础。而多 Agent 系统的社会性使得基于协商的跨站点资源交易成为可能。每个 Agent 代表其所在站点，以所在站点的利益最大化为目标，使用该站点提供的协商策略与其它 Agent 进行交互。

2.1 基于 Agent 的分布式资源库

图 1 给出了基于多 Agent 的分布式资源库架构，其中有静态 agent 和移动 agent。

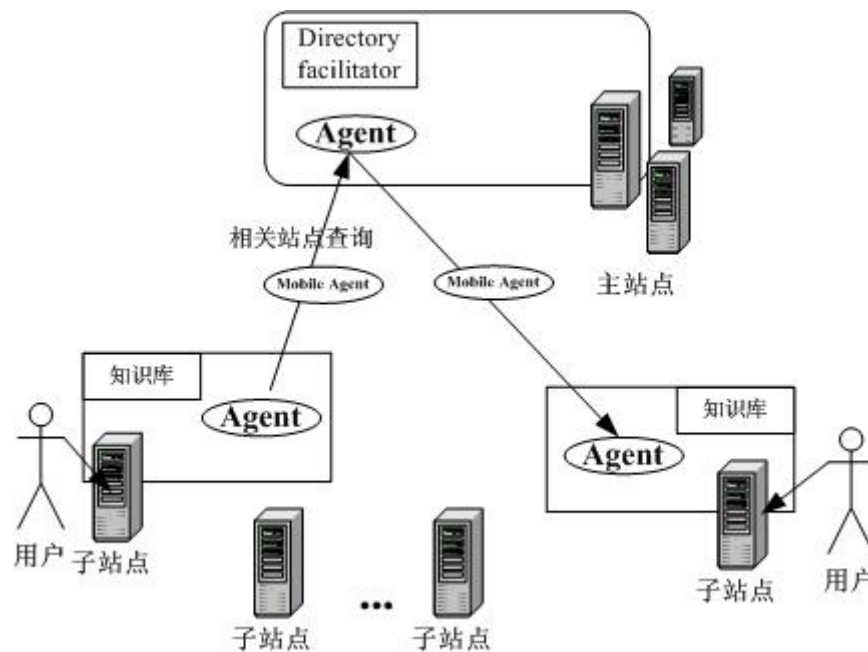


图 1 基于多 Agent 系统的分布式资源库

在这个架构中，每个站点（包括主站点）都由一个静态 Agent 作为核心部件，具体负责如下操作：

- (1) 站点的通信代理。负责代表各个资源站点进行交互，如交换查询请求及结果；负责消息的缓存，管理消息队列。
- (2) 子站点 Agent 通过使用基于本体和刻面的查询得到用户索取资源的案例集，通过案例集在知识库中推出获得用户索取资源的协商信息，将这些协商信息交由移动 Agent 去与其它资源站点交互。
- (3) 主站点 Agent 负责维护和更新子站点目录和历史查询记录，并负责向子站点提供目录服务和子站点列表。
- (4) 负责维护知识库，每当通过协商从其它资源站点获得一个资源后都在知识库中增加一条这次协商中本站点使用到的协商信息。

2.2 移动 Agent

所谓移动 Agent 是运行于开放网络中的封装良好的计算实体，它可代表用户完成特定的任务，具有自主性，移动性，协作性，安全性和智能性等特性。可以携带自身的程序、数据

和状态一起移动，代表源节点在目标主机上运行直至得到最终结果并返回，可以大大降低分布式应用之间的由于中间结果交互而带来的负载和通信开销，是为解决复杂、动态、分布式智能应用而提出的一种全新的计算手段，因此在本系统我们使用移动Agent来完成和多个子站点的协商。

在我们的分布式资源库架构中，移动 Agent 用于完成对索取资源的查询购买。如果资源子站 P1 希望购买某一资源，则该子站点的 Agent A1 动态生成一个移动 Agent M1。M1 携带有从 P1 站点的知识库中获得的协商信息。M1 首先到达主站点，向主站点 Agent G1 提供查询请求，G1 返回一个拥有该资源的子站点列表，M1 根据列表上的子站点顺序来决定自己的路由。当 M1 移动到子站点 P2 时，根据自己携带的 P1 站点的协商模型来与 P2 的 agent A2 交互，经过反复协商后如果得到可以接受协商结果则返回站点 P1，如果 M1 与 P2 无法达成协商目标，则 M1 移动到列表上的下一个子站点继续协商。在移动过程中，M1 还可根据实际情况来决定和调整路由。如果 M1 按照站点列表的顺序到达子站 P2，发现该站点比较繁忙（如 A2 的查询请求队列中未决的查询到达一个阈值），则 M1 可以选择移动到列表上的下一个子站点 P3，以后在适当的时候再回到 P2。

2.3 基于策略的 Agent 协商

对于基于策略的多 Agent 间的协商，其目的是在协商过程中达到自身利益的最大化。其形式类似于我们日常的购买活动，即“货比三家”和选择性价比最好的商品。为了使 Agent 能智能的完成这一过程，我们使用了文献[3]中提到的启发式的自动协商模型。该协商模型将协商分为协商目标和协商策略两大部分，其中，协商目标即为要达成协议所覆盖的问题范围，即协商的问题数量以及对这些问题的底线。协商策略是通过产生一系列报价来实现协商目标。协商模型即为结合了协商目标和协商战略，确定在实际操作中如何报价、反报价及判断协商成功与否的一套机制。下面我们首先给出产生协商模型所要考虑的几个方面：

- (1) 协商包含多个问题（如价格、质量及到货日期），协商成功的标志是所有问题的结果被双方接受。
- (2) 协商双方对协商中每一个问题可以接受的底线在协商前决定的。
- (3) 索取的资源可以由多个站点提供时，能提供该资源的站点个数会对协商结果产生影响。
- (4) 时间对协商是有影响的，一方面，协商双方达成协议所花费的时间应该是合理的，另一方面，双方价格变化的幅度能体现出时间的推移。

以上四点中，协商的问题个数、对每一个问题的底线及协商时间的划定都是从知识库中通过案例推导得出的。而协商的站点个数是由主站点 Agent 提供的。下面我们将解决分布式资源库中不同站点间基于 Agent 的协商，本文采用文献[4]中提到的协商理论来建立协商模型，以一个双边多问题协商为基础，并根据本系统的特点作了改进。协商模型涉及三个方面，分别是：

- (1) 报价评估与报价过程。报价评估的基础是效用函数，效用函数是指对任何一个报价方案，方案中涉及的某一问题 j ，其不同取值对 $Agent_i$ 的效用是不同的。对于一个

$$\text{报价方案 } x = (x_1, x_2, \dots, x_n), \text{ 其效用为 } V^i(x) = \sum_{1 \leq j \leq n} \omega_j^i V_j^i(x_j), \text{ 其中 } \sum_{j=1}^n \omega_j = 1, \omega_j$$

为协商参与者 $Agent_i$ 对不同问题的偏好，这个值的设定对于买卖双方往往是不同的，其选定是由知识库的案例推导得出的。报价评估所要完成的就是在接收到对方的报价方案后，将由对方报价生成的效用函数 $V_{b \rightarrow a}^i(x_j)$ 与己方下一次报价生成的效

用函数 $V_{a \rightarrow b}^i(x_{j+1})$ 进行比较, 如果 $V_{b \rightarrow a}^i(x_j) \geq V_{a \rightarrow b}^i(x_{j+1})$, 则接受对方报价, 否则提出己方的反报价。如果时间到则终止协商。

- (2) 报价策略。报价策略是指根据己方拥有的资源情况而生成下一次报价。这里资源主要包括两方面: 协商对手的个数和协商剩余时间。协商对手多, 则己方选择余地大, 做出的让步就小, 反之做出的让步就大; 关于协商剩余时间, 协商剩余时间越少, 所报出的价格就越接近己方的价格底线。下面我们给出这两者的形式化描述:

- ① 首先给出使用了策略的报价公式: $x_{a \rightarrow b}^t[j] = \min_j^a + \alpha_j^a(t)(\max_j^a - \min_j^a)$, 其

中, \min_j^a 为协商过程中产生过的最低价格, 这个价格一般是购买方所提出的第一次报价; \max_j^a 为协商过程中产生过的最高价格, 这个价格一般是出售方所提出的第一次报价。 $\alpha_j^a(t)$ 为因子函数, 它可以设定为不同形式的函数, 还可以设定参数来控制其随时间变化的程度。它即可以是关于时间的因子函数, 也可以是关于协商对手个数的因子函数。

- ② 时间因子函数 $\alpha_j^a(t) = k_j^a + (1 - k_j^a) \left[\frac{\min(t, t_{\max})}{t_{\max}} \right]^{\frac{1}{\beta}}$ 是文献[4]中提出的, 本文

所采用的就是这个时间因子函数。其中, k_j^a 和 β 都是在开始报价前由知识库的案例推理得出。这种时间因子函数可以通过控制 β 为协商者提供无限种时间策略。当 $\beta < 1$ 时, 协商的策略是保守的, 直到最后才报出保留价格; 而当 $\beta > 1$ 时, 协商的策略是激进的, 在协商开始就报出自己可以接受的最终方案, 而后再也不做让步。两种方式的协商趋势都是相同的, β 越大越激进, 越小越保守。 k_j^a 是让步幅度, k_j^a 越大表明让步幅度越大, 反之越小。

- ③ 本文中使用到的协商对手个数因子函数为 $\alpha_j^a(t) = k_j^a + (1 - k_j^a) e^{-resource(t)}$,

$resource(t) = \mu^a \frac{|N^a(t)|}{\sum_i |x_{i \rightarrow a}^t|}$ 。其中, μ^a 为 $Agent_a$ 认为与单个 Agent 协商的合

理时间花费, 这个也是通过知识库的案例推理得出的。 $|N^a(t)|$ 代表当前时间与 Agent 协商的 $Agent_i$ 的数量, 这个参数的获得是在向主站点 Agent 发出对资源的请求后, 由主站点返回的子站点列表中的站点个数。而 $|x_{i \rightarrow a}^t|$ 表示的是在 t 时刻 $Agent_a$ 与 $Agent_i$ 协商回合长度, 则分母表示的是某时刻 $Agent_i$ 与

$Agent_a$ 协商的总长度。

(3) 协商模型。有了报价策略后，协商战略的目标是如何将报价策略进行组合产生报价

公式。其形式为 $x_{a \rightarrow b}^t[j] = \sum_{i=1}^n \gamma_{ji} T_i^t[j]$ ，其中， $T_i^t[j]$ 表示 $Agent_a$ 在第 j 次报价中

的第 i 种报价策略； γ_{ji} 是在第 j 次报价中对于第 i 种报价策略的权值，且 $\sum_{i=1}^n \gamma_{ji} = 1$ 。

其中 γ_{ji} 也是由对知识库的案例推理得出。最终形成的协商战略函数就是协商模型。

通过以上对协商模型各个构成部分的分析，我们可以看出，协商的成功与否是由基于效用函数的报价评估完成的。协商中的报价与反报价是由一些称为协商策略的函数产生，或者是由这些函数的线性组合来产生。这些策略函数是根据不同的衡量标准(或称为协商影响因素)构建的，如在协商中时间是最重要的影响因素，就构建一个基于时间的策略函数。如果认为协商对手个数也是一个重要的影响因素，同样构建一个基于对手个数的策略函数，这些简单函数在线性组合时可以有不同的权重，体现了不同影响因素的不同重要性。这种策略权重的调整组合就构成了 Agent 的协商战略函数。而由协商战略函数就形成了协商模型。移动 Agent 在协商过程中就可以使用这种协商模型完成对资源的选择和购买。

3. 基于本体与刻面的案例推导

为了构造第 2 节中提到的协商模型，我们需要得到协商信息，例如协商中对价格底线的划定、同每一个资源站点交互时间的设定，不同协商策略在协商模型中的权值比重等信息都要在 Agent 开始谈判前决定下来。为此，我们通过从知识库获得的同这次协商相似的案例并对这些案例进行加权来得出协商信息。本系统中使用了基于本体与刻面结合的方法完成案例推导。下面对这一过程详细介绍。

3.1 本体

本体(Ontology)最早是一个哲学上的概念，是客观存在的一个系统的解释或说明，关心的是客观现实的抽象本质。引入到计算机中后，对本体最为流行的定义是[5]：本体是共享概念模型的明确的形式化规范说明。在计算机中引入本体的目标是[6]：捕获相关领域的知识，提供对该领域知识的共同理解，确定该领域内共同认可的词汇（概念），并从不同层次的形式化模型上给出这些词汇和词汇间相互关系的明确定义。关系表达了领域中不同概念之间的语义联系，例如整体-部分关系、继承关系、属性关系等或其他领域特定的关系。本体如果用相应的逻辑语言来表达，那么还可以具有逻辑推理能力。利用本体中显式存在的概念、实例及关系的知识，通过运用已有的公理，可以推导出许多有意义的隐式知识。如在中学语文这个特定的领域，对于《长恨歌》，其作者是白居易；而白居易的生活时代是唐，由此我们可以推出《长恨歌》的历史背景是唐。

本体建立的概念是在某个特定领域中是公认的，是面向特定领域的概念模型。OWL(Web Ontology Language)[7]作为 W3C 制定的本体语言，它能够清晰的表达词汇表中词条的含义以及这些词条之间的关系，而这种对词条及其关系的表达正是建立并使用本体的基础。因而，我们选用 OWL 来建立本体，在 OWL 中，概念主要由类和实例组成。概念间的关系通过子类关系、属性关系等关系得以描述。还以中学语文这个领域为例，如果有“诗歌”这个类，那么《长恨歌》就是“诗歌”的一个实例；“诗歌”与“时代背景”有属性关系，而“诗歌”

与“绝句”又有一个子类关系。

3.2 剖面

在计算机领域，目前剖面主要用于在软件工程中构件的描述与检索[8, 9]，在剖面的描述方案中，对资源的描述由剖面 and 术语组成。剖面是对资源的一个描述角度。剖面按照组成关系构成一棵剖面树，如图 2 中的“原文信息”和“资源信息”构成顶级剖面。兄弟剖面之间是正交关系，分别表示了对于父剖面来说不同的描述角度；同时又是互补关系，综合起来才能获得较为完整的描述。剖面上的描述值被称为术语，同一剖面下的多个术语之间按照继承关系同样构成一个树形结构，图 2 中剖面“格式”下即为一棵术语树。子剖面与父剖面是组成关系，而子术语与父术语是特殊和一般关系。

基于剖面的描述方法为复杂的资源提供了一种多角度、统一的描述方案，提高了描述的精确性，并具有较高的检索效率。相应的检索过程一般为：首先，将用户的查询要求转化为一棵查询剖面树，然后将此查询树与库中资源的剖面描述进行对比，如果找到一种允许的映射方案，那么该资源即为满足查询要求的一个查询结果。使检索转化为查询树与库中剖面描述树之间的匹配问题。再通过对匹配度阈值的设定，可以选出满足给定匹配度的查询结果。

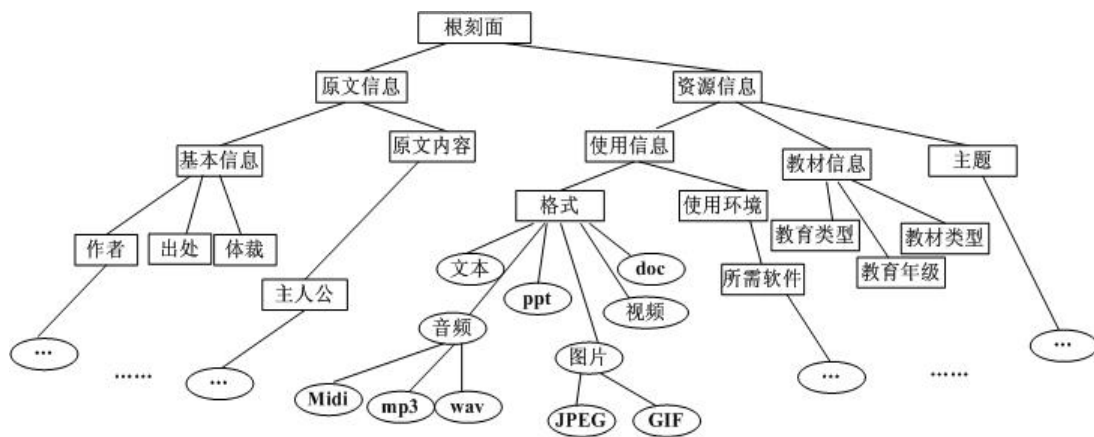


图 2 剖面树

3.3 将剖面与本体相结合

本文引入本体和剖面是为了构造和领域相关的案例库。纯粹的剖面描述和检索只能提供有限的领域知识，因为剖面树和术语树体现的也只是一部份领域知识。在这种描述和检索方法中，用户使用剖面树来表达检索请求时，必须准确完整的给出检索条件，这在很多情况下这是不方便的，仍然以中学语文领域为例，如当用户希望检索标题为“《卖炭翁》”，格式为“.txt”的资源时，在情形一下假设在资源站点上存在“《卖炭翁》.txt”、“《卖炭翁》.doc”、“《卖炭翁》.ppt”三个资源，那么基于剖面的检索可能得到如“《卖炭翁》.txt”这样一个资源。如果在情形二下该站点上只存有“《卖炭翁》.doc”和“《卖炭翁》.ppt”，那么就不能返回任何资源。

如果将本体与剖面的描述和检索方案结合起来，那么就可以为用户提供更强的资源检索服务。资源库中的资源仍然按照剖面方案进行描述，不同的是领域本体建立起了领域内的概念及概念间的关系体系，通过运用显示知识和已有的公理，可以推导出许多有意义的隐式知识，继续上面的例子，在本体库中将图 1 剖面树上的概念“格式”上建立一个“兼容”关系并按照实际将.txt与 doc及 ppt 定义为兼容关系，当我们在情形二下用同样的条件进行查询时我们会查到“《卖炭翁》.doc”、“《卖炭翁》.ppt”。通过将剖面与本体结合后，查询后得到的是一个按照预定的规则进行了智能扩展的资源结果集，而且还可以为该资源集中的每一

个资源都附带一个其与索取资源的匹配度信息。

3.4 知识库与案例推导

本文提到的 Agent 的协商系统具有学习能力, 为了支持 Agent 的这种高级特性, 我们构建了知识库, 每一个资源站点都有自己特定的一个知识库, 其中记录了该资源站点上所有从其它站点取得的资源的协商信息, 即在每次成功的从其它站点购买了一个资源后, 都要在这个知识库中记录在购买这个资源的过程中本站点所使用到的协商信息。

结合从 3.3 中取得的资源结果集, 可以从知识库中获得所有取得这些资源时的协商信息, 这些资源的协商信息即为可用于这次协商的案例集, 以 3.3 中得到的这些资源结果集中的资源与本次索取资源的匹配度为权值, 对案例集中的每一案例的协商信息进行加权得到这次协商中将使用的协商信息。

对于案例推导, 基于刻面与本体结合的查询所拥有的一个优点是通可以过已有知识对新知识进行推理, 仍以 3.3 中的情形二为例, 当我们希望购买一个“《卖炭翁》.txt”时, 在本站点的资源库中我们找不到这样一个教学资源, 即这是一个从未出现过的新资源, 我们无法得知其价格, 也不知道为了获得这个资源该使用什么样的协商信息。但是在建立了本体后, 我们就可以知道“《卖炭翁》.doc”、“《卖炭翁》.ppt”与我们所要购买的“《卖炭翁》.txt”是有关系的, 我们可以根据这两个已有资源的协商信息推导出新资源“《卖炭翁》.txt”的协商信息。再结合查询得到的匹配度, 我们就可以得到一个基于某个特定站点所包含的知识及其知识容量的协商信息。有了这些协商信息之后, Agent 就可以按照第 2 节提到的协商策略进行资源的比较和购买。

4. 购买方 Agent 的交易实例

这里我们以购买方 Agent 的一次购买过程为例, 说明基于 Agent 协商的分布式资源库的工作机制。

4.1 协商信息的获得

我们使用图 1 所示的刻面树作为资源站点 R 1 对其所拥有资源的描述和检索方法, 在本例中使用到的本体中的关系有:

1. 兼容关系, 即由文件后缀表明的文件格式间的兼容, 如.txt、.doc、.ppt 两两之间存在兼容关系。
2. 文章与作者间的隶属关系, 如《卖炭翁》、《长恨歌》都与白居易存在“隶属”关系。
3. 当一篇文章是对另一篇文章或是对某一个人的评论、简介时, 这篇文章与其评论、简介的对象之间存在“评论对象”这样一个关系, 如《<卖炭翁>的社会背景和历史意义》与《卖炭翁》间存在“评论对象”关系, 《白居易生平》与白居易间也存在“评论对象”关系。

在此次交易中, 购买方资源库请求的资源是: “《长恨歌》.doc”, 它首先通过本体与刻面相结合的查询方法从其自身的资源库中得到与查询目标相关的资源列表。如图 3 所示

Resource Selector			
File			
Resource		Facet Association	
id	title	medium	matching
400102	《卖炭翁》	格式-txt	0.50
400214	《长恨歌》	格式-ppt	0.75
401045	《<卖炭翁>的社会背景和历史意义》	格式-txt	0.25
401261	《白居易生平》	格式-txt	0.50

图 3 资源查询结果列表

对于查询返回的资源列表，将在知识库中取得获得这些资源时的协商信息作为案例集，对案例集中的案例按照资源列表中的匹配度信息进行推导，得到资源“《长恨歌》.doc”的协商信息，如图 4 所示，这张表所示信息为：此次交易中该资源站点所能接受的最高报价为 25¥，在第一报价中己方的报价为 15¥（这里以购买资源的站点为例，所以其价格底线位最高价，而第一次报价为最低价。对于提供资源的站点将相反），协商模型中使用的两个策略函数（时间策略函数与个数策略函数）的权值为 0.4 和 0.6，在时间策略函数中， β 取值为 3.5， k_j^a 取值为 0.2，对于个数策略函数，同任何一个资源站点进行协商的最大时间为 500ms。而谈判对手的个数信息（即拥有该资源的站点的个数）需从主 Agent 那里取得。

Negotiation Info	
Highest Price	25¥
Lowest Price	15¥
Time Tactic Modulus	0.4
B in Time	3.5
K in Time	0.2
Num Tactic Modulus	0.6
Negotiation Time	500ms
Quit	

图 4 协商信息参数表

4.2 使用协商模型进行交易的过程

接着资源请求站点将创建一个携带有协商信息的移动 Agent M1，M1 首先从主 Agent 那里取得拥有资源“《长恨歌》.doc”的资源站点列表，站点个数即为表示协商中协商对手个数的 $|N^a(t)|$ 。然后动态的移动到那些站点上（假设它移动到了站点 P2 上），使用其携带的协商信息构建协商模型进行协商，在第 2 节中我们给出了协商过程的原理，这里我们以一个协商回合来说明协商模型是如何工作的。

首先得到站点 P2 对资源的报价为：30¥，M1 得到这个报价后求出该报价的效用函数，这里，我们为了简化描述，假设协商目标中的协商问题只有价格，所以效用函数

$V^i(x) = \sum_{1 \leq j \leq n} \omega_j^i V_j^i(x_j)$ 的取值等同于价格的取值，且有 $V_{P2 \rightarrow M1}^{M1}(1) = 30 \text{ ¥} > V_{M1 \rightarrow P2}^{M1}(1) = 15$

¥，（如图 4 所示，15¥为 M1 第一次将报出的价格），由效用函数看出，P2 此次报价的效用值高于 M1 下次报价的效用值，所以 M1 将给出反报价 15¥，之后 M1 使用协商模型计算自己下次的报价，其具体过程为先分别由时间策略函数和个数策略函数得到新报价为 20¥和 15¥，再结合协商模型中的策略权值得到下次报价为 $20 \text{ ¥} * 0.4 + 15 \text{ ¥} * 0.6 = 17 \text{ ¥}$ ，即 $V_{M1 \rightarrow P2}^{M1}(2) = 17 \text{ ¥}$ ；在 M1 计算下次报价的同时，P2 收到 M1 的第一次反报价后，根据它自己的效用函数进行计算，给出新报价 $V_{P2 \rightarrow M1}^{M1}(2) = 25 \text{ ¥}$ ，这里我们可用清楚的看到，双方的报价都向有利于对方的方向进行了调整，这与现实中人与人之间的协商交易相同。M1 在第二次得到 P2 的报价后，与自己的效用函数比较得到 $V_{P2 \rightarrow M1}^{M1}(2) = 25 \text{ ¥} > V_{M1 \rightarrow P2}^{M1}(2) = 17 \text{ ¥}$ ，即 P2 此次报价的效用值仍高于 M1 下次报价的效用值，此时 M1 与 P2 谈判花费了 200ms，小于最大谈判时间 500ms，所以 M1 给出自己新的反报价 17¥，与 P2 继续进行新回合的谈判。

双方通过效用函数判断对方的报价是否可以接受，直到某一方的报价被另一方接受，协商成功结束；或者是协商超过了某一方的最大协商时间，则协商以失败告终。如果在某一站点协商失败，M1 仍可以移动到其它站点继续交易，直到取得自己需要的资源或与所有站点的协商都失败。当两个站点成功协商后，双方将都在自己站点的知识库中记录此次交易中的协商信息。这样，随着交易次数的增多，知识库的容量将不断变大，该站点的协商模型会不断改进。

5. 总结

本文的目标是解决分布式资源库下不同站点间资源交易的自动化。整个系统建立在多 Agent 架构之上，不同站点间的资源交易由代表该站点的 Agent 来完成。引入本体和刻面后我们就可以从资源库中查找到与所需资源性质相近的那些资源构成的案例集，使用案例集从知识库中获得案例的协商信息并根据匹配度权值进行案例推导，进而得到在购买资源时协商模型所需的协商信息，最终由携带了协商模型的 Agent 代表所在资源站点完成交易。系统实现了新商品交易中价格的判定及协商模型的选取，同时随着站点交易次数增多，知识库容量增大，协商模型将越来越有效。

今后的工作主要集中在两个方面：

- (5) 在构造案例集阶段，进一步完善本体与刻面结合的查询策略，使得案例集的选取更为合理，在新商品交易中推导出更为有效的协商信息。
- (6) 对于目前的协商模型，改进已有的关于价格的协商策略以提高其效率，同时引入对协商目标中除价格之外的其它问题（如质量、交货时间等）的协商策略，实现能够解决多问题的协商模型。

参考文献：

- [1] Michael Wooldridge 著，石纯一 张伟 徐晋晖 等译。多 Agent 系统引论，2003-10，15-16 页
- [2] JADE 官方网站 www.jade.com
- [3] 高璐，毛波，王方，基于 MAS 的企业供应链模型中谈判过程的研究，系统工程理论方法应用，第

11 卷第 4 期 2002 年 12 月

[4] Carles Sierra, Peyman Faratin, Nick R. Jennings, A Service-Oriented Negotiation Model between Autonomous Agents, Proc 8th European Workshop on modeling Autonomats Agents in Multi-Agent World[C]. Sweden, 1997. 17

[5] Studer R, Benjamins V R, Fensel D. Knowledge Engineering, Principles and Methods. Data and Knowledge Engineering, 1998, 25(122): 161~197

[6] 邓志鸿, 唐世渭, 张铭, 杨冬青, 陈捷, Ontology 研究综述, 北京大学学报(自然科学版), 第 38 卷, 第 5 期, 2002 年 9 月

[7] W3C 关于 OWL 的官方网址: <http://www.w3.org/TR/owl-features/>

[8] 马亮, 谢冰, 杨芙清, 多构件库统一剖面检索机制, 电子学报, Vol.30 No.12A Dec. 2002

[9] 王渊峰, 张涌, 任洪敏, 等. 基于剖面描述的构件检索. 软件学报, 2002, 13(8): 1546~1551