

基于对话的多 Agent 协作模型研究

邓璐娟, 陈培, 潘凯洁

(郑州轻工业学院 计算机与通信工程学院, 河南 郑州 450000)

摘要:针对传统协作设计过程中的盲目性和随意性问题,提出一种基于对话的多 Agent 协作模型.该模型首先利用“与/或”结构任务树的形式来简化任务,再通过划分通信区域建立区域 Agent 服务器,以提高通信效率;再依据对话模型定义 Agent 对话交互语义,并将其应用于多 Agent 的协作过程.仿真结果表明,该模型能够为多 Agent 协作提供一种灵活、有效的交互手段,并能够显著提高系统的运行效率.

关键词:任务分解; Agent 通信;对话语义;协作模型;交互手段

中图分类号:TP311.4 **文献标志码:**A **DOI:**10.3969/j.issn.2095-476X.2012.06.002

Study on the cooperation model of the multi-Agent based on dialogue

DENG Lu-juan, CHEN Pei, PAN Kai-jie

(College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China)

Abstract: Aiming at the blindness and randomness of the traditional cooperation in design process, a dialogue-based multi-Agent cooperation model was proposed. The task was simplified by the task tree form of "and/or" structure. Communication efficiency has been raised through dividing the communications area and establishing Agent server. Agent dialogue interaction semantics were defined according to the dialogue model and used in multi-Agent cooperation process. The simulation results showed that the model which is Agent cooperation design provides a flexible and effective means of interaction, and the efficiency of the system can be significantly improved.

Key words: task decomposition; Agent communication; dialogue semantics; cooperation model; interactive tools

0 引言

多 Agent 系统提供了一种解决复杂问题的分而治之的方法.传统的多 Agent 协作设计过程中,每个 Agent 对于所要完成的任务拥有不全面的信息和能力,无法根据协同任务中其他 Agent 的行为需求来调整优化自身行为,从而影响协作效率和系统的灵活性^[1].本文针对传统的多 Agent 协作设计过程中

的盲目性和随意性,提出一种基于对话的多 Agent 协作模型,以期在一定程度上实现多 Agent 之间的理性交互,为多 Agent 协作提供一种灵活、有效的交互手段.

1 任务分解

把问题不断地分解和细化是解决复杂问题的重要方法.对于任务耦合度较大的 Agent,建立一种

收稿日期:2012-09-14

基金项目:国家自然科学基金项目(61040025)

作者简介:邓璐娟(1964—),女,湖南省浏阳市人,郑州轻工业学院教授,博士,主要研究方向为控制理论和控制工程.

“与/或”结构来描述任务分解是一种有效的方法. 根据这种层次的树状结构, 规定树的根节点为总请求任务, 树中同层节点间具有“与/或”关系. 若节点 T 有 n 条边分别链接子节点 T_1, T_2, \dots, T_n , 且这 n 条边为逻辑“与”的关系, 则需要多个 Agent 协作完成任务 T , 即 $T = T_1 \wedge T_2 \wedge \dots \wedge T_n$; 若 n 条边为逻辑“或”的关系, 则只需要其中的某个 Agent 就可以独立完成任务 T , 即 $T = T_1 \vee T_2 \vee \dots \vee T_n$. 采用这种方式将任务逐层地划分下去, 一直划分到满足最小子任务的条件为止, 子任务之间也存在“与/或”关系. 根据这种逻辑关系, 对每个分支通过目标函数 M 进行综合开销运算, 即

$$M = \sum_i \sum_j Z_{ij} \text{ExecFun}(A_i, E_j) + \sum_i \sum_j W_{ij} \text{Commfun}(E_i, E_j)$$

式中, $\text{ExecFun}(A_i, E_j)$ 为 Agent j 操作 i 的执行开销, $\text{Commfun}(E_i, E_j)$ 为 2 个 Agent 间的通信开销, Z_{ij} 和 W_{ij} 分别代表 $\text{ExecFun}(A_i, E_j)$ 和 $\text{Commfun}(E_i, E_j)$ 在综合开销运算中所占权重. 根据 M 算出每个分支的综合开销, 保留最小消耗分支去除其他分支并删除该“或”分支的根节点, 可完成对该任务树的修剪, 达到简化任务的目的.

2 多 Agent 通信模式

多 Agent 通信模式是: 建立一个 Agent 通信区, 将交互程度紧密的 Agent 通过总的区域 Agent 置于同一个通信区内, 共同信息以黑板系统的方式经过共享通道传递给区域内成员.

符号名是各个 Agent 识别的标识. 为了保证一致性, 采用统一的 Agent 命名机制, 将区域管理 Agent 命名为管理 Agent(MA), MA 的职责是管理区内其他 Agent 的名字, 维护相关的信息(如 Agent 的专业领域、角色类型、完成任务的能力大小和当前活动的状态等). 新加入系统的 Agent, 必须先向 MA 进行名称注册, 注册成功后的 Agent 具有同域中其他 Agent 进行通信的权利^[2]. MA 也会及时记录各个 Agent 的能力或其他参数的变化情况. 当任务发生时, 请求 Agent 通过询问 MA 来选择相应的通信对象 Agent 来承担任务, 此时可以将 MA 理解为通信服务器. 在原始的通信模式中, 每个 Agent 需要维护一张各个 Agent 的地址和功能表, 不仅耗费通信资源, 而且效率也不高. 相比较而言, 通过 MA 的方式来完成 Agent 之间的寻址和定位的方式更加灵活有效.

3 基于对话的多 Agent 协作模型

3.1 基于对话的多 Agent 协作模型设计

对话模型分为信息搜索对话、查询对话、劝说对话、协商对话、慎思对话 5 种语言^[3]. 一次对话至少发生在 2 个 Agent 之间, 说话者 Agent 需要选择正确的语言和语法有计划地表达自己的意图, 听话者 Agent 通过分析说话者的言语行为来准确理解其意图, 它不仅关心自身的状态和行为, 还要关注其他 Agent, 最终在服务于总体任务的条件下, 根据说话者 Agent 的建议修改自身的愿望和意图.

在对话过程中, 用 T 表示协作请求任务, Agent A 为任务的发起者, 通常 A 即为通信模型中的 MA, $B = \{b_1, b_2, \dots, b_n\}$ 表示目前整个协作系统中的 Agent 集合.

基于对话的多 Agent 协作过程可分为以下 3 步.

第 1 步: 任务发起者 A 依次与任务承担着 b_i 建立信息搜索对话, 询问其专业领域及参与完成任务 T 的能力、意愿、机会, 以确定 b_i 能否成功参与协作并完成任务. 查询过程如下:

A 查询 b_i 专业领域的征询式为

$$\text{req}_{A, b_i}(\text{specific of } b_i, t)$$

b_i 根据自身状态知识库并结合对话策略做出合理的应答, 即

$$\text{statement}_{b_i, A}(\text{"Vehicle Enigeering"}, t + 1)$$

根据 b_i 的回答, A 能够初步判断 b_i 能否承担起任务 T . 若 b_i 具有这样的能力, 则 A 进行对话; 否则结束对话, 继续寻找下一个合适的服务 Agent.

A 询问 b_i 承担任务 T 并提供服务的愿望征询式为

$$\text{req}_{A, b_i}(\text{Accept}(b_i, T), t + 2)$$

b_i 对于承担 T 的意愿有 2 种可能, 即 $\text{Accept}(b_i, T)$ (b_i 愿意承担任务 T) 和 $\text{Refuse}(b_i, T)$ (b_i 不愿意承担任务 T). 因此, b_i 可做出如下 2 种应答:

$$1) \text{prom}_{b_i, A}(\text{Accept}(b_i, T), t + 3)$$

$$2) \text{prom}_{b_i, A}(\text{Refuse}(b_i, T), t + 3)$$

基于 b_i 的应答, 任务发起者 A 就会根据“if trust, then believe”的规则更新其社会模型知识库, 最终形成一个愿意共同完成协同设计任务的潜在 Agent 集合 G .

第 2 步: 根据信息搜索对话的结果, A 选用劝说对话, 依次劝说 G 把实现请求任务 T 作为其意图, 也就是说 G 中每个 Agent 均把实现任务 T 作为它们各

自的意图. 劝说过程如下:

当 b_i 做出的应答为 Refuse (b_i, T) 时, 往往还会给出拒绝的原因, 即

$$\text{inf}_{or_{b_i,A}}(\text{"reason"}, t + 4)$$

A 对 b_i 的拒绝原因给予考虑, 并再次提出改进建议后的请求, 即

$$\text{prom}_{A,b_i}(\text{Accept}(b_i T'), t + 5)$$

若 b_i 能够接受此建议, 则向 A 做出回应

$$\text{prom}_{b_i,A}(\text{Accept}(b_i, T'), t + 6)$$

或者 b_i 也可根据自身的情况对 A 的建议提出反建议

$$\text{req}_{b_i,A}(\text{Counter-Advice}(A, T''), t + 7)$$

最后 A 接受 b_i 的反建议, 协商成功, 做出回应

$$\text{prom}_{A,b_i}(\text{Accept}(A, T''), t + 8)$$

第3步: 当 G 中的所有成员都拥有了上述意图, A 将以联合信念的形式予以确认, 并广播到每个成员, 最终形成联合意图, 即各个 Agent 间能够成功进行合作. 到此, 此次协作过程结束.

3.2 基于对话的多 Agent 协作过程效用分析

不确定性和不可靠性一直是多 Agent 间协作的重要障碍. 例如完成某项总任务需要 5 个 Agent 共同协作, 然而潜在的协作 Agent 集合 $H = \{20\}$, 根据随机事件规律, 恰好选中的这 5 个 Agent 的概率低于 0.01, 这样选择的盲目性很可能导致执行任务结果异常. 针对这一问题, 采用对话协作模型的优势显而易见.

根据每个预分解设计子任务, A 采取广播方式对 b_i 的专业领域进行并行查询, 进而了解到该查询对象 Agent 的相关参数, 包括完成子任务的能力、兴趣和机会. 在满足专业领域匹配的条件下, 再比较兴趣和机会的概率值^[4], 若两者也满足某给定值, 那么能力值高的 Agent 优先被选择. 由于对话带有时间属性, 因此若对象 Agent 在规定时间内没有做出响应则会被认为是拒绝请求, 放弃任务. 一旦确定选择 A, 它将会从后续任务的备选集合中被剔除. 在这种对话模式下, 任务分解分配的盲目性和随机性可以得到有效控制.

由于 Agent 的专业领域、能力、兴趣和机会具有差异性, 如何选择并确定最合适的 Agent 是衡量协作效率的另一个重要指标^[5]. 然而任何的协作交互都需要占用系统资源, 造成通信开销, 能否正确地对 Agent 间的协作效用与开销进行合理的计算分析, 就显得非常重要. 多 Agent 协作效用与开销的计算式对话协商效用评价模型为

$$(A, B, P_B, P_{A,B})$$

其中, A 表示 Agent A, 即对话协商发起者; B 表示 Agent B, 即对话应答者; P_B 表示对话中 Agent B 所需占用的系统资源, 即所需要付出的代价; $P_{A,B}$ 表示对话协商中 Agent A 认为 Agent B 所需要付出的代价, 即理论上系统认为可以为此次对话分配的系统资源. 将代价 P 简化成维持一个 Agent 的代价 C_1 , 建立对话的代价 C_2 , 进行对话协商的代价 C_3 . C_1 的权重为 ξ_1 , C_2 和 C_3 的权重均为 ξ_2 , 则得到

$$P_B = \xi_1 C_1 + \xi_2 (C_2 + C_3)$$

$$P_{A,B} = \xi_1 C'_1 + \xi_2 (C'_2 + C'_3)$$

其中, $C_1 = \sum_{i=1}^{\lambda} \theta_i P_i t$, θ_i 表示第 i 个 Agent 的状态, 活动状态 $\theta_i = 1$, 挂起状态 $\theta_i = 0$; t 是活动的时间. λ 个 Agent 的协作效用 $F(\lambda) = \sum_{i=1}^{\lambda} P_{A,i}$, 该协作效用等价于协作中所有 Agent 的代价值之和.

对话协商的效用分析可以定义为

$$\begin{cases} F(\lambda) = \sum_{i=1}^{\lambda} P_{A,i} \\ F(\lambda) \leq \sum_{i=1}^{\lambda} P_i & \text{拒绝 Agent } i \text{ 加入协作集合} \\ F(\lambda) > \sum_{i=1}^{\lambda} P_i & \text{接受 Agent } i \text{ 加入协作集合} \end{cases}$$

综上所述, 管理 Agent 需要对新加入的 Agent 进行效用和代价的评估: 希望协作效用大于协作代价, 否则协作的效用全部用来维持集合中各个 Agent 的代价开销.

4 仿真试验

下面以室内温度调节控制为例, 进行仿真实验. 在不采取任何措施的情况下, 若房间中的热负荷变化, 则温度也会相应地发生改变. 当房间实际温度与设定温度差超过人体舒适度要求时, 房间 Agent 就会向管理 Agent (MA) 发出室温调整任务请求, 并将温差参数传递给 MA. MA 接收到任务请求后, 根据知识库中的知识, 通过调节风机转速和送风温度这 2 个子任务即可完成温度调节任务, 而各子任务又需要若干个子任务以并发或是串行的方式来完成^[6]. 因此, 可根据“与/或”结构任务树的划分规则进行任务分解.

图 1 为温度调节任务的一个协调过程, 在芝加哥大学社会科学计算研究中心研制的 multi-Agent 建模工具 Repast 软件平台上实现了此过程中各个

Agent 的协作交互. 经过系统仿真, 试验结果如图 2 所示.

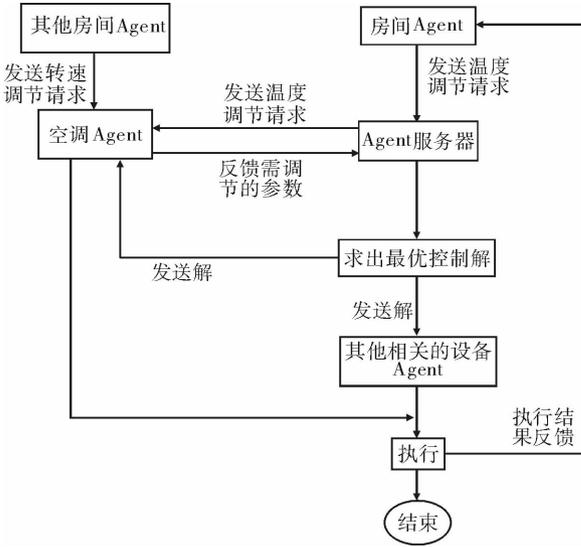


图 1 温度调节协调过程

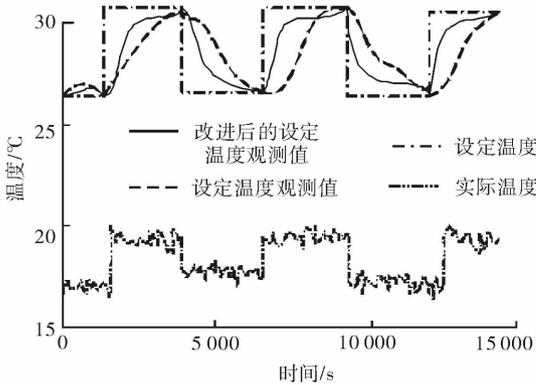


图 2 温度变化参数曲线

从图 2 可以看出, 采用对话语义的 Agent 协作交互能够有效提高 MAS 控制的时效性和准确性, 送风温度可以更快地达到设定值, 性能明显优于一般的 Agent 协作模型. 图 3 为在 Repast 软件平台上, 当任务书由 10 个增加至 80 个时, 通信代价测试结果. 试验结果证明, 随着任务数的增多, 基于对话语义的 Agent 协作模型的通信代价大大低于普通的协作方式, 节省了时间, 提高了运行效率.

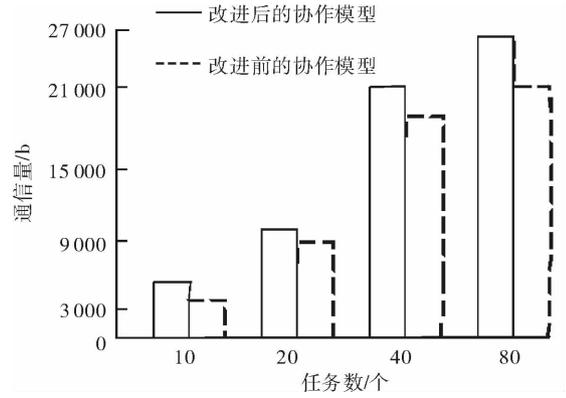


图 3 通信代价测试结果

5 结论

本文提出了一种基于对话的多 Agent 协作模型: 基于 Agent 协作模型中“与/或”结构任务树任务分解模式, 采用划分通信区域、建立区域 Agent 的通信模式, 给出了基于对话交互语义的对话交互过程. 通过该模型, 不仅将完成一个独立复杂的任务转变成多个 Agent 的联合意图, 使多个 Agent 各自的行为具有相关性、一致性, 而且使得任务的分配充分考虑各辅助 Agent 的专业领域、能力、兴趣和机会. 试验结果表明, 通过对话协作模式大大提高了多 Agent 间相互协作的有效性和灵活性.

参考文献:

- [1] 陈志. 基于 Agent 的无线传感器网络若干问题研究 [J]. 南京邮电大学学报, 2007, 27(3): 216.
- [2] 谢学科. 多 Agent 交互协作研究及系统模拟 [D]. 西安: 西北工业大学, 2005.
- [3] 安毅生, 李人厚. 基于对话的多 Agent 协作交互模型 [J]. 西安交通大学学报, 2005, 40(12): 1344.
- [4] 孟建良, 孔维莉, 庞春江, 等. 基于系统体系结构的多 Agent 协作 [J]. 微机发展, 2005, 15(12): 73.
- [5] 杨爱琴, 朱玲玲, 程学云. 基于流演算的多 Agent 请求/服务协作模型的研究 [J]. 计算机工程与设计, 2011, 32(2): 681.
- [6] 马琴. 基于多 Agent 的智能建筑集成管理平台的研究 [D]. 上海: 同济大学, 2009.