

基于MAS的群决策支持系统研究

徐振宁, 张维明, 陈文伟

(国防科技大学管理科学与工程系, 长沙 410073)

摘要: 群决策支持系统(GDSS)与传统的决策支持系统(DSS)在体系结构和工作方式上有着较大的差异。在网络环境中,组织的决策需要一种开放的GDSS体系结构和协作的工作方式。本文提出一种基于MAS的GDSS体系结构,以及在这种体系结构下的协同决策的模型。在此基础上开发了一个GDSS的原型系统,并对今后的研究方向进行了展望。

关键词: 多主体系统; 群决策支持系统; 协作; 体系结构; 决策模型

中图分类号: TP18; TP316 **文献标识码:** A **文章编号:** 1007-9807(2002)01-0085-08

0 引言

随着计算机软硬件,并行计算,分布处理技术研究不断取得新的进展,决策支持系统(decision support system, DSS)也在不断地与新技术、新学科相互交叉融合、发展完善。处理节点计算和通信能力的不断增强,促使组织中出现了大量能够收集和整理至关重要信息的位置^[1],支持和协调复杂组织群体问题求解情况的DSS变得越来越重要。为此,研究者将支持群体问题求解的决策支持系统看作是群决策支持系统(group decision support system, GDSS)。

从DSS用户的角度看,DSS没有得到更为广泛应用的主要原因包括^[2]:不知道相关决策支持技术的存在;由于可移植性等问题难以获得这些技术;将已存在的技术与特定问题匹配,应用到特定问题过程中的困难;交互式决策模型要求的数据集成代价昂贵等。

从DSS开发者的角度看,Spriegel^[3]于1980年提出了具有三个组成部分的DSS结构,它们分别是数据库部件(data base, DB)、模型库部件(model base, MB),以及将两者联系起来的用户接口部件(user interface, UI)。为了克服三部件结

构难以反映出人类决策行为的特性,1985年,Belew把数据、文本、模型、规则分离出来,把以规则形式存在的人类经验知识作为DSS的一个独立的部件^[4],使DSS结构更清晰,行为特征更能反映人类的决策过程。随着计算机软硬件,并行计算,分布处理技术研究不断取得新的进展,复杂组织对支持和协调群体协作问题求解的DSS的需求日益增加,新的需求促使现有DSS的结构作进一步的发展^[5-7]。

从80年代末开始,最早源于分布式人工智能(distributed artificial intelligence, DAI)的多主体系统(multiagent system, MAS)理论与技术从DAI领域中拓展出来,在许多不同于最初DAI的应用领域中得到了更广泛的应用。本文提出一种基于MAS的GDSS结构以及GDSS协同决策模型,并对今后的研究方向进行展望。

1 基于MAS的GDSS结构

对GDSS的需求源于两个相互冲突的组织情况^[8]:对组织中信息共享更大程度的需求,以作为应对环境复杂性和不确定性的一种手段;同时希望简化制定决策的agent群体内的协调。通过创

建基于MAS的GDSS,组织中的成员能够透明地获取组织中其他成员的知识技能,通过向系统中加入具有新功能的agent或增加现有种类agent的数量,或者agent通过自身的学习机制,能够动态地实现系统功能的扩充,能够同时处理多个不同的决策任务,满足用户对信息不断变化的需求,克服现有DSS系统体系结构的难以扩充、相对封闭的不足

1.1 Agent原理和MAS

1.1.1 Agent的一般原理

研究者从不同的角度和目标出发对术语“agent”赋予了不同的含义。Wooldridge和Jennings对“agent”的不同定义进行了总结,给出了目前基本被学术界公认的定义^[9]。术语agent的弱定义是将其定义为具有如下特性的计算机软件或硬件系统:自主性(autonomy)、社会性(social ability)、反应性(reactivity)、能动性(proactivity)、时间连续性(temporal continuity)以及面向目标性(goal orientedness)。符合强定义的agent除了具有弱定义中的特征外,还具有以下一种或多种特性:可移动性(mobility)、理性(rationality)、适应性(adaptivity)、协作性(collaboration)。

Agent体系结构主要研究软件或硬件如何实现agent的方式。在实现一个agent时,传统的方式是将其作为一个基于知识的系统,这种主体的体系结构被称为“慎思的”(deliberative architectures)。与之相对的有一种不使用符号知识的agent体系结构,被称为“反应的”(reactive architectures),还有一种综合了两者长处的agent体系结构,被称为“混合的”(hybrid architectures)。

很多agent理论研究者希望在逻辑的框架下对具有意向性的概念进行形式推理,其中最具有代表性的理论模型被称为信念(belief)—愿望(desire)—意图(intention)模型,或BDI模型。在这种理论框架下,意图、信念、目标等高层认知功能被看作(模态)算子,通常采用可能世界语义学的方法解决语义问题。

1.1.2 MAS的基本原理

DAI中的MAS技术将人类智能的本质看作是一种社会性的智能,人类绝大部分的活动都涉及多个人构成的社会团体,大型复杂问题的求解需要多个专业人员或组织协作完成。MAS放松了

对集中式、规划、顺序控制的限制,提供了分散控制、应急和并行处理,并且MAS可以降低软件或硬件的费用,提供更快速的问题求解。

目前在MAS的研究中,往往运用博弈论、经济学、社会学等方法刻画和控制自私agent的行为。另外一种往往是基于BDI模型对系统中agent的行为进行规范和描述。Agent之间的协作和协调是多主体研究的核心问题之一,协作是非对抗的主体之间保持行为协调的一个特例。

1.2 基于MAS的GDSS体系结构

在本文中研究的agent是大粒度、慎思、半自治松散耦合的agent。大粒度是指GDSS中的每个agent都具有相当的问题求解能力,拥有一个或多个进程的资源。松散耦合是指agent之间是协作关系,相互依赖的程度较低,没有集中的控制中心,但是每个agent都要受到社会规则的约束,还要对承诺守信,所以又不是完全自治的agent。慎思是指agent具有高层认知状态,根据对高层认知状态的推理,决定自身的行为。因此,GDSS中的agent应当采用BDI结构的模型。

1.2.1 GDSS中agent的基本结构

在众多现有的BDI结构模型中,GRATE^{*}(generic rules and agent model testbed environment)第一次将联合意向和联合责任的概念引入到BDI结构中,从而使BDI agent的研究进入MAS的阶段。当然GRATE^{*}也存在这样一些问题^[10],在基础性的描述性理论和体系结构之间,在功能和实现的体系结构之间存在一定的脱节。GRATE^{*}的体系结构没有涉及小组形成过程的细节,社会规划的产生以及角色的分配等。这些问题的处理留给应用的设计者根据组织的策略和agent对其它agent能力的信念,定义相应的机制。

系统借鉴GRATE^{*}的模型结构,给出了一种GDSS中agent的一般功能结构,如图1所示。该结构试图体现通过agent遵守社会规则,隐式地体现联合意向和愿望,这是与GRATE^{*}结构最大的不同。现有的BDI体系结构都没有如何将描述中令人满意地由愿望形成意图的过程体现出来,这可能是因为愿望对不同的设计者有着不同的语义^[10]。系统认为在同一组织中所有的agent都遵守将组织利益的最大化作为自身的愿望,都在认为协作能够促使组织利益最大化时发出协作请

求, 都在认为协作能够促使组织利益最大化时对协作请求做出承诺的社会规则 这样 agent 在有能力处理协作请求时, 不会出现由于自身考虑而拒绝请求的情况 由于 agent 知识的不完整性, 仍将出现局部利益和全局利益的冲突, 不过这也反映了组织中不同部门考虑利益的不同角度 这可以通过为决策者提供多个决策方案, 由决策者根

据整体利益来协调不同 agent 间的利益冲突, 或者采用能够体现决策者协调策略的效用函数, 解决冲突 这样每个 agent 的愿望和联合意图就隐式地体现在 agent 所要遵守的社会规则中, 他们的根本愿望是一致的 Agent 的高层认知状态推理主要集中于如何利用自身有限的资源和能力满足系统利益最大化的需求

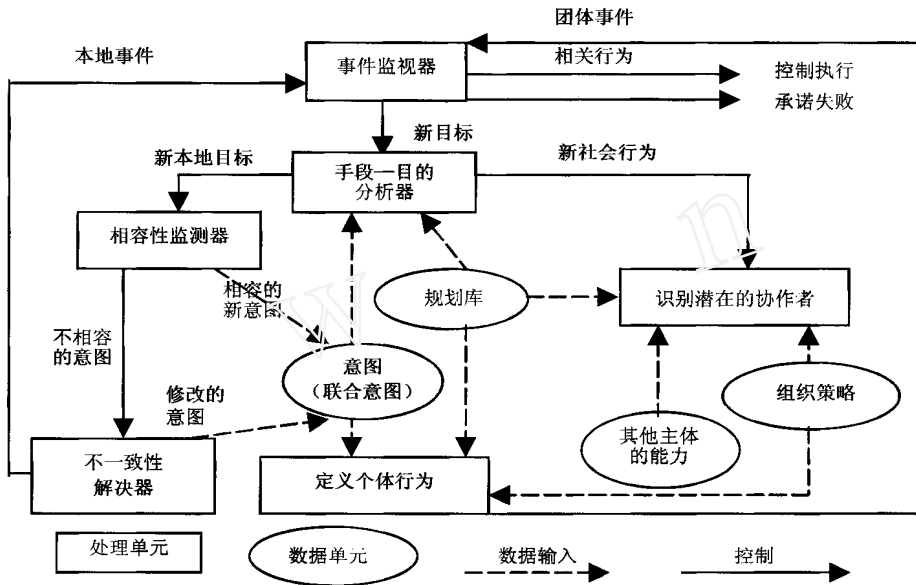


图 1 GDSS 中 agent 的一般功能结构

在功能结构图中, 事件监视器监视本地问题求解结果, 环境变化和组织中其他 agent 发生的事件等对 agent 有意义的事件 事件意味着对新行为的潜在需求, 意味着可能产生新的目标 新目标作为手段—目的分析器的输入 当有新目标需要分析时, 手段—目的分析器参考规划库寻找能够完成目标相应的规划 这些规划表明目标是在本地实现, 还是需要协作实现 同时, 手段—目的分析器根据社会规则和已经存在的意图, 比较 agent 已经承诺的行为与新目标的重要性 如果手段—目的分析器决定在本地实现目标, 相容性监测器要确保新目标以及实现新目标的行为与已经存在的意图相容 如果相容性监测器监测的结果不一致, 不一致性解决器尝试修改已经存在的承诺, 或者修改(包括放弃)新目标来消除冲突 通常, 不一致性解决器的输出将导致手段—目的分析器一系列进一步的分析

如果手段—目的分析器决定协作完成目标,

必须建立一个社会行为 首先根据对其他 agent 能力的认识和组织策略识别团体中哪些 agent 是潜在的协作者 然后根据相应的规划和组织策略, 将规划中需要完成的各项任务和潜在的合作伙伴进行匹配, 将各项任务传递给相应的协作者 最终, 各协作者各自执行手段—目的分析, 相容性监测, 各 agent 间循环往复, 直到协作任务完成, 或失败

1.2.2 GDSS 的体系结构

网络环境中的群决策支持系统包括 5 种类型的 agent, 根据 GDSS 的应用规模, 系统中可以同时存在多个同种类型的 agent 实例 这 5 种类型的 agent 分别是人机接口主体 (human computer interface agent, HCIA), 模型管理主体 (model management agent, MMA), 知识管理和学习主体 (knowledge management & learning agent, KMLA), 信息服务主体 (information service agent, ISA), 协作服务主体 (cooperation service agent, CSA). 同一组织中的 agent 通过知识通信

总线(know ledge communication bus, KCB)进行交互,如图2所示这种 agent 类型的划分,借鉴和继承了传统DSS 体系结构的组成要素,同时结合了MAS 的技术,各类 agent 根据自身的知识完成了协作角色的划分。

各 agent 具有统一的模型结构,拥有其它 agent 的能力模型,遵循统一的工作流程,同时根据自身功能的需要对结构中不同组成部分的侧重有所不同 GDSS 中不同种类的 agent 由事件驱动,协作完成组织中的各项任务。

HCA 是人机交互的接口,具有传统DSS 用户接口部件的功能,而且是决策规划产生者、决策规划协作执行的发起者、决策规划执行的监督者和决策方案的提交者 即HCA 根据自身的知识负责问题求解临时小组的形成,利用规划库中的

规划产生社会规划

MMA 是模型管理者,具有传统DSS 的模型管理功能 模型集成是按照建模任务,根据建模规则,不断的将子模型集成、调整,直至生成符合最终目标的集成模型 模型集成最基本、最核心的思想就是经过一系列规范的操作把一些标准的模型进行有机集成,生成满足一定标准的集成模型 可见,具有一定数量的领域模型和规范的模型集成操作是模型集成的前提条件 系统按说明部分、连接部分、参数部分和模型体 4 个部分对模型进行规范和建模 MMA 根据其规划库中针对不同问题的模型生成、组合的规划,动态的生成、组合以及控制模型的运行,并在解决新问题后将新规划加入到规划库中

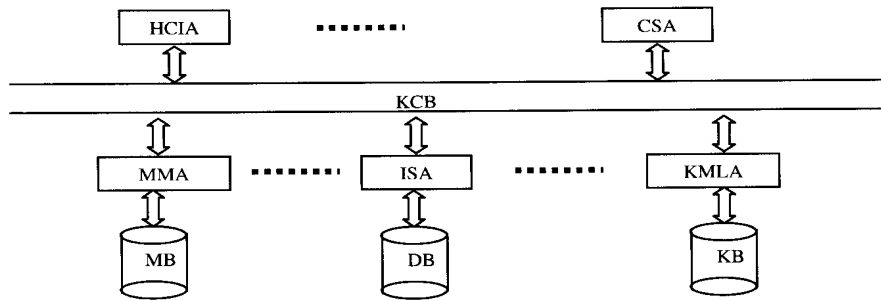


图2 基于MAS的GDSS结构图

自从Belew 把以规则形式存在的人类经验知识作为DSS 的一个独立部件,DSS 的研究者逐渐对DSS 系统应该结合数值计算和符号推理达成了共识 KMLA 具有自己的决策和求解范围,并且拥有与之相应的自己的知识库(know ledge base, KB)和问题求解方式,主要承担决策任务中符号推理的任务,完成传统专家系统的功能

ISA 具有传统DSS 数据库部件的功能,同时ISA 具有主动的数据服务功能 这主要体现在当ISA 自身不能满足其它 agent 数据准备请求时,能够主动地与其它 ISA 协作,透明地向其它agent 提供数据服务;能够为其它 agent 主动提供持续的数据服务,例如能够为其它 agent 提供数据预定(subscribe)等服务

CSA 在GDSS 的体系结构中发挥着重要的协调作用 由于每个 agent 不可能完全掌握环境和其他 agent 的信息,每个 agent 掌握的信息也不可能完全一致,因此 agent 在遇到自身无法解

决的问题时并不意味着 GDSS 系统无法解决该问题,这也正是MAS 技术要解决的问题之一 每个 agent 按一定的规则向 CSA 通知自身能力、状态的变化,这使得某个 CSA 或多个 CSA 能够掌握系统最新的能力和状态,使得某个 agent 自身不能解决问题时,征求整个系统的帮助成为可能

文[11]讨论了仅仅在 agent 通信语言层编程的可能性,指出知识层编程的通信应该满足的条件:异步传输机制、可靠的消息传递、非阻塞的原语等 基于MAS 的GDSS 中的通信是知识层的通信, KCB 采用 KQML (know ledge query and manipulation language) 通信语言和通信机制,支持局域网和广域网的知识层通信

2 GDSS 协作决策模型

基于MAS 的GDSS 系统中存在多种协作方式,其中最主要的方式就是知识和能力共享的协

作方式,即多个 agent 协作完成同一个决策方案,在此基础上同一个决策问题存在多个方案时,agent 根据自身在系统中的角色同时处理多个方案中的相关部分,最终由决策者协调采用哪种方案

当组织中的所有 agent 都遵守如上所述的社会规则时,对 agent 间协作需求的产生、协作关系的建立、协作过程的进行和协作的终止等发生条件的判断和 agent 具有的高层认知状态推理就可以转化为对 agent 获取组织中技能和知识能力的计算。组织中 agent 之间协作关系的形成,通常是由于协作 agent 之间知识和能力的结合能够完成单个 agent 无法或难于解决的决策问题。因此,本文从每个 agent 协作所需知识的角度,形式化地定义协作行为的发生、进行和终止

协作系统中的知识被看作是一个 5 元组,即 $K = \{C, d, P, AU, OI\}$, 其中 C 为常识 (common sense), 表示所有 agent 都遵守的社会规则和处理问题的方式等知识; d 为领域知识 (domain knowledge), 表示 agent 具有的领域技能, 例如 MMA 具有模型表示、生成、组合、运行等领域知识; P 为外部知识 (peripheral knowledge), 它是交互知识的重要组成部分, 表示 agent 需要协作时能够向谁、获取什么帮助; AU 为权限知识 (authority knowledge), 表示 agent 愿意将自身知识和能力提供给那些 agent, 提供到什么程度; OI 为其它的交互性知识, 例如语言知识等

定义 1 $O = \{A_1, \dots, A_n\}$ GDSS 被看作是 agent 的集合, A_i 表示 agent $_i$, O 表示 GDSS 组织

定义 2 $C = K_1 \dots K_n$, K_i 表示第 i 个 agent 拥有的全部知识, C 表示所有 agent 拥有的共同知识

定义 3 $D_0 = \{d_1, \dots, d_n\}$, d_i 表示第 i 个 agent 拥有的问题求解相关领域知识和技能, $d_i = \{d_{i1}, \dots, d_{im}\}$, d_{ij} 表示第 i 个 agent 拥有的第 j 条领

域知识和技能, D_0 表示组织整体拥有的与问题求解相关的领域知识

定义 4 $P_i = \{(d_{xy}, A_j, \xi_i) \mid \forall d_{xy} \in d_i, d_{xy} \in N_i, \forall A_j \in O, j \neq i\}$, P_i 表示 agent $_i$ 拥有的外部知识集合, 外部知识包括关于其它 agent, 及其它 agent 的知识和技能的知识, N_i 表示 agent $_i$ 自身不具有, 但是解决特定问题时所需的与问题求解相关的知识集合, A_j 表示向哪个 agent 请求帮助, ξ 表示 agent $_i$ 从 agent $_j$ 获取知识或技能 d_{xy} 可能性的大小

需要说明的是可能有不止一个 agent 拥有 d_{xy} , agent 也可能知道有某个 agent 拥有 d_{xy} , 但是不能确定是哪一个 agent

定义 5 $d_{ij} \in d_i$, 如果 A_k 能够访问某个 d_{ij} , 则 $au_{d_{ij}}^k = 1$

$AU_{d_{ij}} = \{A_k \mid au_{d_{ij}}^k = 1, \forall A_k \in O, k \neq i\}$, $AU_{d_{ij}}$ 表示能够访问某个 d_{ij} 的 agent 的集合

$AU_i = \{AU_{d_{ij}} \mid \forall d_{ij} \in d_i\}$, AU_i 表示能够访问 agent $_i$ 各项知识和技能的 agent 集合的集合

定义 6 协作需求的发生条件为 agent $_i$ 自身没有进行问题求解所需的知识 d_{xy} , 即 $d_{xy} \notin d_i$, 但相信系统中存在这一知识

定义 7 协作关系的建立: 根据协作请求类型的不同, agent 间协作关系的建立分为直接和间接协作两种方式

I 直接方式: 如果协作发起者请求非 CSA 类型的 agent 协作, 请求建立协作关系的通信行为 r_j^i 表示如下:

$r_j^i: (d_{jx}, A_j, \xi_j) \in Act_{d_{jx}}$, 如果 $(d_{jx}, A_j, \xi_j) \in P_i$, i 表示发出请求的 agent, j 表示接收到请求的 agent, d_{jx} 表示被请求的知识, 作为行为标记的一部分, 没有特殊含义

对于 r_j^i 的应答 r_j^i 表示为

$$r_j^i = \begin{cases} d_{jx} & \text{如果 } A_i \in AU_{d_{jx}}, \text{ capacity}_j(\text{now}), AU_{d_{jx}} \in AU_j \\ \emptyset & \text{如果 } A_i \notin AU_{d_{jx}}, AU_{d_{jx}} \in AU_j \\ B & \text{如果 } A_i \in AU_{d_{jx}}, \neg \text{capacity}_j(\text{now}), AU_{d_{jx}} \in AU_j \\ u & \text{如果 } d_{jx} \in d_j \end{cases}$$

capacity $_j$ (now) 表示被请求的 agent $_j$ 当前具有处理该协作请求的能力, B 表示 busy, 暂时不能

处理

II 间接方式: 如果协作发起者请求 CSA 类型

的 agent 协作, 协作关系建立的通信行为 r_k^i 或 $r_k^{i'}$ 表示如下: i 表示发出请求的 agent, k 表示接收到请求的 CSA 类型的 agent, j 表示拥有知识 d_{jx} 的 agent:

$$r_k^{i'} = \begin{cases} (d_{jx}, A_j, \xi) & \text{如果 } A_i \in AU_{d_{kx}} \text{ capacity}_k(\text{now}), AU_{d_{kx}} \in AU_k \\ \emptyset & \text{如果 } A_i \notin AU_{d_{kx}}, AU_{d_{kx}} \in AU_k \\ B & \text{如果 } A_i \in AU_{d_{kx}} \rightarrow \text{capacity}_j(\text{now}), AU_{d_{kx}} \in AU_k \end{cases}$$

d_{kx} 表示关于如何直接获取知识 d_{jx} 的知识, $\text{know}_k(d_{jx} \in A_j)$ 表示 agent k 知道哪个 agent j 拥有 d_{jx} , A_i 接收应答后的行为及相应的应答同定义 7

$$r_k^{i'} = \begin{cases} (d_{kx}, A_k, \xi) & \text{如果 } A_i \in AU_{d_{kx}} \text{ capacity}_k(\text{now}), AU_{d_{kx}} \in AU_k \\ \emptyset & \text{如果 } A_i \notin AU_{d_{kx}}, AU_{d_{kx}} \in AU_k \\ B & \text{如果 } A_i \in AU_{d_{kx}} \rightarrow \text{capacity}_k(\text{now}), AU_{d_{kx}} \in AU_k \end{cases}$$

其中 d_{kx} 表示关于间接获取知识 d_{jx} 的知识, $\text{know}_k(\text{who_know}_k(d_{jx} \in A_j))$ 表示本 agent k 知道哪个 agent k' 知道谁拥有 d_{jx} , A_i 接收到应答后的行为同定义 7 中的 (*), 相应的应答同定义 7 II 的

否则, 应答表示为

$$r_k^{i'} = u$$

定义 8 A_i 建立协作关系的步骤表示如下:

- I 确定协作者集合, $A \text{ Set}_j = \{A_j | d_{xy} = d_{xy}(d_{xy}, A_j, \xi) \in P_i\}$, agent i 收集到的可能拥有 d_{xy} 的协作者, “= ”表示 d_{xy} 与 d_{xy} 等效
- II 如果 $A \text{ Set}_j \neq \emptyset$, 从 $A \text{ Set}_j$ 中依次按 $(d_{xy}, A_j, \xi) \in \xi$ 中的大小顺序选择 A_j 请求协作
- III 如果 $A \text{ Set}_j = \emptyset$ 或第 II 步的请求全部失败, 则依次从存在于 A_i 外部知识中的 CSA 类型的 agent 集合 $\{CSA_p, \dots, CSA_q\}$ 选择 CSA' 请求协作

如上定义, agent 通过交互性知识 P, AU, OI 和常识 C , 支持 GDSS 系统的协作求解; 通过组织的领域知识 D_0 , 对特定问题进行求解

3 应用实例

下面介绍结合军事中的情报处理, 实现的一个多主体协同智能军事态势评估系统原型^[12], 验证如上的概念定义. 系统中应用了基于知识系统

$r_k^i: (d_{kx}, A_k, \xi) \in AU_{d_{kx}}$, 如果 $(d_{kx}, A_k, \xi) \in P_i$ (*)
如果 $\text{know}_k(d_{jx} \in A_j)$, $r_k^{i'}$ 的应答 $r_k^{i'}$ 表示如下:

中的 I.
如果 $\text{know}_k(\text{who_know}_k(d_{jx} \in A_j))$, $r_k^{i'}$ 的应答 $r_k^{i'}$ 表示如下:

中广泛使用的 KQML 通信平台. 该平台为系统中的主体提供了知识级通信的手段, 具有应用无关性, 独立于具体的应用背景, 有力地支持了系统的扩展性

系统中共有五类主体, 它们分别是协调控主体 (head agent, HA), 目标识别主体 (project judge agent, PJA), 用户交互主体 (user interact agent, UIA), 共享信息服务主体 (share information service agent, SISA), 实时信息输入主体 (real-time information agent, RIA). 参考 BDI 主体结构, 系统实现了单个主体的框架结构. 根据主体在协作求解问题中拥有的知识及知识处理能力进行协作角色划分, 由问题求解所需的知识和知识处理驱动, 形成协作关系

每个主体维护一个自身问题求解的领域 ontology (本体库), 包含主体问题领域的概念和概念之间关系的定义. 每个主体的领域知识和外部知识分别应用自身的 ontology 进行表示. HA 中的 ontology 包含整个问题域的概念和概念之间关系的定义, 作为不同 ontology 交互的基础, 确保不同主体对同一概念定义的一致性. 以 SISA 为例, 介绍 ontology 的定义 (如图 3), 外部知识和权限知识的表示与应用

SISA 负责为其它主体提供信息服务, 它必须主动为其它主体搜集自身也没有掌握的信息. 例如, SISA 并不知道在某一具体时刻, 某个地点是否为敌方的防御重心, 上级部门对某个地点目标

或态势的判断, 即 Location 的属性 important 和 superior 的值, 对这些外部知识的表示为:

Concept Hierarchy	Attribute Definitions
Object[] Location Object Coordinate Location Target Object Tank Target Command - Post Target -----	Location[name= >> STR NG; flatness= >> STR NG; highland= >> STR NG; coordinate = >> Coordinate; important= >> STR NG; superior= >> STR NG; ----- } Coordinate[x= >> NUM ERAL; y= >> NUM ERAL; }
Rules	----
FORALL Target1, Target2 Target1: Target[Location coordinate- >> X, Time- >> timeA] <- Target2: Target[Location coordinate- >> X, Time- >> timeA] -----	

图 3 SISA ontology 的部分实例

U IA Location (

Location important: STR NG,
Location superior: STR NG,
-----)

即对 U IA 掌握的关于 Location 知识的描述

U IA 针对 SISA 访问 Location 知识的权限知识表示为 +, +, ..., 即 SISA 具有访问 Location 知识的前两个属性的权限

参考文献

- [1] W ginston A. Intelligent agents as a basis for decision support system s[J] Decision Support System s, 1997, 20(1): 1-1
- [2] Tung X. Bui Decision support in the future tense[J] Decision Support System s, 1997, 19(3): 149-150
- [3] Sprague R H A. Framework for the Development DSS[J] M IS Quarterly, 1980, 4(4): 1-25
- [4] 陈文伟 决策支持系统及其开发[M] 北京: 清华大学出版社, 2000 311-326
- [5] 向 阳, 黄梯云 基于管理问题理解的 DSS 智能构模理论框架研究[J] 管理科学学报, 1999, 2(3): 51-58
- [6] 孙 波, 林宣雄, 李怀祖 领导辅助决策支持系统研究[J] 管理科学学报, 1998, 1(3): 65-74
- [7] 彭俊松, 黄丽华, 薛华成 基于知识共享重用的分布式 DSS 开发平台[J] 管理科学学报, 1999, 2(1): 72-78
- [8] DeSamctis G, Gallupe B. A foundation for the study of group decision support system s[J] M anagement Science, 1987, 33(5): 589-606
- [9] Wooldridge M, Jennings N R. Intelligent agents: theory and practice[J]. Knowledge Engineering Review, 1995, 10(2): 115-152
- [10] Afsane H, Kurt S. Belief-desire-intention agent architectures[A] Foundation of distributed artificial intelligence [M] New York: John Wiley & Sons Inc, 1996 169-186
- [11] Mauro G. Concurrency and knowledge-level communication in agent languages[J]. Artificial Intelligence, 1998,

当 PJA 请求 SISA 提供信息服务时, SISA 将自身拥有的某个地点的信息以及应用如上描述的外部知识主动向 U IA 搜集的信息, 统一提供给 PJA, 供 PJA 进行目标识别 即 PJA 与 SISA, SISA 与 U IA 分别形成协作关系, 共同完成目标识别

4 结束语

本文在对现有 DSS 体系结构进行深入分析的基础上, 依据用户的需求和相关技术的最新进展, 提出一种基于 MAS 的 GDSS 体系结构, 给出了单个 agent 的一般功能结构图和构成系统的 agent 种类, 在此基础上, 从 agent 协作所需知识的角度, 形式化地定义协作发生、进行和终止, 给出了 GDSS 协作的模型, 并在实践中对模型进行了检验

基于 MAS 的 GDSS 是一种全新的 DSS 的体系结构, 有待于在实际的问题中进一步的应用、检验和完善 作为一种新的体系结构, 无论是 MAS 还是 GDSS 都给研究者提出了新的问题, 如涉及 MAS 学习方法的 KMLA 学习, 采用 agent 技术实现模型的自动组合生成, 以及问题求解过程中的冲突消解问题等, 都有待于进一步的研究

105(1/2): 1-45

- [12] 徐振宁, 姚 莉, 张维明等. 基于MAS 的协作信息系统试验平台的设计与实现[J]. 计算机工程与应用, 2000, 36(7): 31-34

Study of GDSS based on MAS

XU Zhen-ning, ZHANG Wei-ming, CHEN Wen-wei

Department of Management Science and Engineering, The National University of Defense Technology, Changsha, 410073, China

Abstract GDSS has great difference with traditional DSS on working style and architecture. In the network environment, decision in organization needs a kind of open GDSS architecture and cooperation working style. This paper presents a new GDSS architecture based on MAS and a cooperation decision model under the architecture. Based on architecture, we develop a prototype of GDSS and figure out the direction of research in the future.

Key words: multi-agent system (MAS); GDSS; cooperation; architecture; decision model

(上接第 58 页)

- [13] 刘海龙, 郑立辉等. 证券投资决策的微分对策方法研究[J]. 系统工程学报, 1999, 14(1): 69-72
- [14] Liu B D, Ku C F. Probability criterion in inventory systems[J]. Journal of Systems Science and Mathematical Science, 1993, 13(1): 70-75
- [15] 梁建峰, 唐万生. 有交易费用的组合证券投资的概率准则模型[J]. 系统工程, 2001, 19(2): 5-10

Problem of two period investment decision-making with probability criterion

HAN Qi-heng, TANG Wan-sheng, LI Guang-quan

Institute of Systems Engineering, Tianjin University, Tianjin 300072, China

Abstract Probability criterion is an investment decision making criterion of investors in some cases, it has practical significance, its investment decision making is determined by the expected return rate. Generally speaking, different investors will have different expected return rates, the security's return rates in different periods will have different probability distributions and correlation to each other. In this paper, a problem of two-period investment decision-making with probability criterion is proposed, its mathematical model of an optimal investment decision-making is established. For continuous or discrete random variable of security's return rate, the steps for investment decision-making are derived. Finally, an illustrative example is given.

Key words: investment decision making; probability criterion; rate of return