

群体决策支持系统 GDSS:模型和分类

Group Decision Support Systems: Models and Classification

吴朝晖 莫振饶 潘云鹤

(浙江大学人工智能研究所 杭州 310027)

摘要 群体决策支持系统 GDSS 作为计算机支持的协同工作(CSCW)的分支,与传统的决策支持系统 DSS 有着较大的差别,前者将面临多个参与者且决策权力是分层的,本文介绍了集中式与分布式群体决策支持系统及其群体决策模型,并展望了未来的工作。

关键词 群体决策支持系统,集中式,分布式,群体决策模型

GDSS

模型 分类

群体决策支持系统(GDSS),指的是“一个基于计算机的系统,通过让一组决策者们以群体的形式一起工作,使非结构化的、难以决策问题更易于解决”;它作为计算机支持的协同工作(CSCW)的分支,是一门较新的、具有巨大发展前途的研究领域。

GDSS是从传统的决策支持系统(DSS)发展而来的,但和 DSS 又有着较大的差别。DSS 的对象是单个用户,且决策具有较高自动化;而 GDSS 面临着多个参与者,并将决策的权力分层次地留给了参与决策者们。

GDSS 主要包含了两方面的技术:一是,包含信息表达和处理技术,例如,相关信息的收集、整理、表示及分析。二是,还需要解决群体协同的计算技术,如通讯,冲突的解决,一致性控制,和数据访问的权

限控制。所以,群体决策所涉及的领域相当广,还涉及社会学,心理学,计算机硬件,软件,网络技术,以及人工智能。

有了 GDSS,人们就能通过计算机举行会议、进行决策,而不受时间和地点的限制。它和人工的会议、决策相比,应该具有如下优点:更能将人们的注意力集中在相应的问题上;能帮助决策者们理解复杂的问题和环境;更能促进决策者之间的相互交流,增进他们之间的了解和信任;能获得更佳的决策;能达成更高的一致程度。

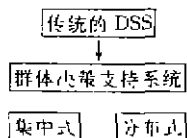


图 1 GDSS 分类

可减少编写程序的工作量。

4. CORBA 支持同一场地和不同场地上的程序间的通讯,因而可将程序的通讯交给 CORBA 完成,减少程序的复杂性。

5. 利用 CORBA 的异常处理功能可方便地处理网络异常和与服务相关的各种异常。

参考文献

- [1] OMG, The Common Object Request Broker: Architecture and Specification, revision 2.0, July, 1995
- [2] A. Dogac et al., A Multidatabase System Implementation On CORBA, 6th Intl. Workshop on Research Issues in Data Eng., Feb. 1996
- [3] J. R. Nicol et al., Object Orientation in Heterogeneous Distributed Computing System, IEEE Computer, 26(6)1993
- [4] Edmond Misrobian et al., OASIS: An Open Architec-

ture Scientific Information System, 同[2]

- [5] OMG, The Object Database Standard, ODMG-93, release 1.2, 1996
- [6] F. Manola et al., Distributed Object Management, Intl. J. of Intelligent and Cooperative Information System, 1(1), World Scientific, 1992
- [7] ObjectBroker, System Integration Guide, Digital Equipment Corp., Aug. 1994
- [8] 王国仁, 基于自治模块和柔性联盟的大规模数据库技术的研究及一个面向对象的多数据库集成系统的设计与实现, 东北大学博士论文, 1995
- [9] P. Evaggelia, B. Omran, Object Orientation in Multidatabase Systems, ACM Computing Surveys, 27(2) 1995
- [10] E. Bertino et al., Integrating Heterogeneous Database Applications through an Object-oriented Interface, Inf. Syst., 14(5)1989

GDSS 的分类如图 1 所示。按照决策的方式, GDSS 可分为集中式和分布式, 它们各自有不同的模型和相关技术。但是, 共同的是, 它们都涉及了多个决策者, 而且决策者之间存在着相互作用、相互影响, 这种相互作用、相互影响是通过信息交流、信息共享来实现的。

在集中式决策模式下的群体共同地解决同一个问题, 其焦点在于引导和协同各个参与者之间的相互影响、相互作用, 使得问题能由群体成员共同同时地解决。在分布式决策中, 这个大问题被分割为一些小问题, 将它们分配给各个子群体, 可以同时解决, 也可以不同时解决。

一、集中式决策

在集中式决策中, 所面对的是一个具有紧密结合的、所有群体成员共同致力于一个共同目标的群体。一个问题的决策通常需要提供这样几个步骤: 问题的确认, 提议解决方案, 讨论, 整理, 和完成决策。

1 三阶段模型

三阶段模型将整个会议进程分为三个阶段: 智力、设计和选择, 每一阶段又包括一个发散性的分阶段和一个集中性的分阶段。在智力阶段完成对问题的分析和定义。群体中产生各种不同的定义, 这就是发散阶段; 而对各种不同的定义进行评价, 在其中的一个定义上取得一致, 这就是集中阶段。在设计阶段中, 先形成对问题的各种可能解决方案(发散阶段), 再将相类似的进行合并, 多余的及无关的删除(集中阶段)。选择阶段包括了对已产生的想法的发散性评估和集中性选择。

该模型基于这样一个基本假定: 只有当足够多的时间花在了讨论问题和提出不同选项上, 才会产生最佳的决策; 产生的不同选项越多, 作出的决策就越好。正由于采纳了这种“越多越好”的前提, 所以用来评价群体相互作用程度的变量, 除了各种选项的质量好坏以外, 还包括其数量多少。

让我们来看一个“纸-笔”会议实例。会议中所有的交流都通过纸和笔来完成。假设共有 N 个成员参加会议, 而事先准备了 $N+1$ 份空白文件。每个成员先在一份文件上发表自己对问题的看法, 完成以后将这份文件抛出, 任换另一份文件, 继续发表对问题的看法或对他人的意见作出评价; 由于共有 $N+1$ 份文件, 所以任何时候都能交换文件, 而不会发生冲突。GDSS 的三阶段模型正是受到了这种“纸-笔”会议技术的启发, 实现了群体成员之间的信息交流。

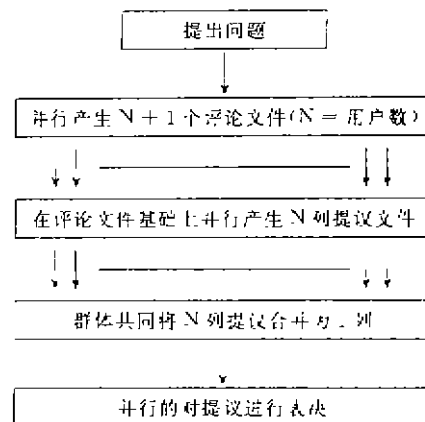


图 2 三阶段群体决策模型

图 2 显示三阶段群体决策模型的结构及数据流图。在这种模式下, 一次会议从一个问题开始, 问题被复制到了 $N+1$ 个文件中 (N = 用户数)。在智力阶段, 用户任意地在 $N+1$ 个文件中发表意见, 由此发展了 $N+1$ 个讨论流。智力阶段之后用户独立工作, 产生 N 列对问题的提议及解决方案。然后由群体将这 N 列合并成一列。最后在这一列上进行表决。

这种模型通过实践, 证明是成功的, 但还存在着一些问题。首先, 不允许用户致力于某个特定文件中的一个特殊问题, 也不允许用户对所有意见作出反应。在一个较大的群体中 ($N > 16$), 对于某些文件, 用户可能只获得一次接触机会, 甚至一次也没有。其次, 有一个独立的提议、解决方案产生阶段, 但不允许用户在表决前进行讨论和修改, 也不允许用户在表决中提出新的方案。第三, 由于各用户独立工作, 在产生大量方案的同时, 也不可避免地引起了重复。将重复消除, 将个人工作成果合并, 这需要一个漫长的过程, 而且个人独立工作时间越长, 合并就越久。

这些经验使我们认为, 向参与者提供更多的并行活动和获取群体最大贡献能力之间存在着基本矛盾。换句话说, 提高个人的贡献能力并不一定会提高群体的贡献能力。由此我们可以定义并行矛盾: 群体的贡献能力小于其成员贡献能力的总和。

决策群体系统 (Decision Making Group Systems) [Nunamaker et al., 1987; Dennis et al., 1988] 采用的就是这种三阶段模型。它由三个组件组成: 电子风暴系统 (Electronic Brainstorming System) 支持智力阶段; 问题组织者 (Issue Organizer) 支持设计阶段; 表决 (Voting) 支持选择阶段。该系统由于结构清晰, 易于使用, 享有较高的知名度, 被广泛地使用。

2 共享上下文模型

前面所提到的三阶段决策模型,由于存在着一些问题,导致我们采用一种新的方法:共享上下文这一个新概念,它指的是一种共同的工作环境,包括信息、信息的表示(如:图像,文本)和对信息的一致理解。

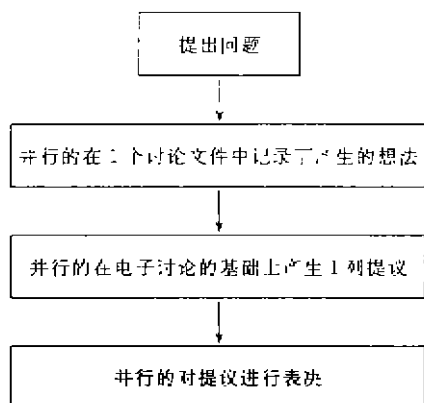


图3 共享上下文群体决策模型

图3说明了采用共享上下文模型的群体决策过程。和图1一样,共享上下文模型从一个问题开始,用户可以自由地发表对问题的看法,提出子问题或相关的注意项。当讨论进入成熟状态后,用户们开始提出解决方案。讨论继续进行,用户对别人提出的解决方案发表看法,提出各种各样的提议。当一轮表决完成之后,方案系列缩短了,而进程继续进行下去。

共享上下文模型是基于这样一个基本假定:加强群体的合作会相应地增进其贡献能力和工作效果。而相反的观点认为:引起冲突(及随后的消除)会增加工作结果的质量。有实验将它和三阶段模型作了比较。结果表明,三阶段模型产生更多不同的解决方案,但共享上下文模型的决策质量更高,一致程度也更大。

共享上下文模型采用了一些技术来处理并行矛盾。首先,它向所有用户同时提供全部数据的访问权限,避免了数据重复,减轻了数据冗长。其次,它将用户的输入区分为一般评论、问题、解决方案,从而产生了更多电子对话的方向和焦点。第三,它在讨论、分析、表决各进程之间快速循环,更利于产生一致方案,达成协议。

由于采用了共享的上下文,对合并阶段的需要消失了,唯一的合并工作是在用户输入时,将新的输入并入共享的上下文。

在用户接口方面,采用了“我见即你见”(What-

You-See-Is-What-I-See, WYSIWIS)的策略,即共享的上下文对所有用户保持一致的显示。共享的上下文数据在各个工作站和网络服务器上被复制,每个工作站均可以修改服务器上的上下文,同时也有责任提醒其它工作站,共享的上下文发生了变化。WYSIWIS是这样实现的:如果一个工作站修改了上下文,a)服务器上的上下文以写的方式打开,修改完毕后文件关闭,并通知各工作站;b)接到修改通知的工作站可以申请直接修改他们本地的文件拷贝,也可以按只读的方式打开服务器上的上下文以作出彻底修改。

此外,手势(Gesture)技术指的是一种指向别人屏幕上某一处的能力,对于群体讨论也很重要。电子指针就是该技术的一种实现。这并不是一项新技术,但以前只为一个群体提供一个电子指针,而现在可以向每个用户提供一个。通过对它的使用,用户可以将别人的注意力集中到相应的问题上来,并通过电子讨论解决问题。

电子讨论系统(EDS)最初是由 Arizona 大学发展起来的,它采用的就是共享上下文模型。使用 EDS 的用户,在屏幕下半部的私人窗口完成一项输入以后,将它归入三类中的一类(评论,问题,提议),并释放给群体。用户可以打开卷动显示模式(Scrolling Mode),这样,任何一个用户的输入都能自动显示在屏幕上半部的公共窗口中。三类输入分别由黑、蓝、红三种颜色表示。

讨论文件是以后交换信息的基本工具。一次 EDS 会议可能有多个讨论文件(文件数和参与者数目无关),允许所有参与者对全部讨论文件进行访问。讨论文件由于会议中群体致力于某个特殊问题而产生。由此,EDS 允许参与者致力于某个问题的各个方面,或同时致力于多个问题。

当讨论进入到成熟状态后,参与者可以提出表决要求。事先可设置一个数字,当提出表决要求的参与者数达到要求时,即进入表决状态,进行表决。

一次会议后,会议中各项输入、表决结果都以文件的形式储存下来,所有活动都标上了时间及相关工作站,以供分析。

二、分布式决策

分布式决策的概念是 80 年代初由 Thomas 和 Burns 等人提出的,和集中式决策相比,最大的差别在于,分布式决策所针对的问题非常复杂,复杂到了单个的参与者无法了解它的全部。

分布式决策活动按照这样的步骤进行:(1)问题分割;(2)任务分配;(3)各自分问题的解决;(4)汇总。当然,实际上很多系统并不支持全部四个过程,认为问题分割和任务分配事先就已经完成,通常只致力于分问题的解决和汇总过程,多群体决策支持系统严格地体现了这一工作进程结构。

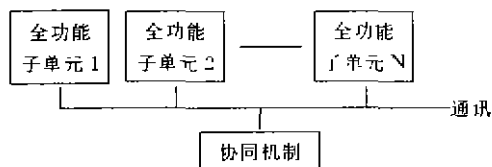


图4 分布式决策的系统结构

进行分布式决策的组织具有多个不同的子单元。这些子单元可以是单个人,也可以是一个群体;可以是同构的,也可以是异构的。他们各自拥有完备的决策支持系统,根据各自的任务、目标和环境提出自己的观点,独立地工作;但是,各子单元的工作却又是相互影响、相互作用的,他们通过一个通讯网络共享各自的观点,协同工作。(见图4)。

各子单元的观点常常存在着矛盾,所以需要一个协同机制,它不仅是单纯的收集各种观点,还要认识到冲突,并通过对话进行讨论以解决冲突。SPIDER系统支持下的观点提取过程较好地解决了这一问题。

此外,由于分布式决策中的多任务,对通讯网络的管理也变得更加复杂和困难,传统的客户机/服务器两层结构显得不足。对此,几位台湾研究学者提出了群件服务器/应用程序服务器/客户机的三层结构。

1 多群体决策支持系统

多群体决策支持系统(MGDSS),帮助多个群体的参与者理解复杂的问题,帮助他们在行动上取得一致来解决这些问题。这些群体松散结合,可以是同构的,也可以是异构的。如图5所示,MGDSS进程主要有三个进程:会前计划,进行会议,会后活动。

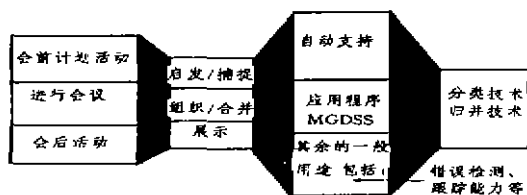


图5 多群体决策支持系统的进程模型和技术支持

会前的详细计划对于会议的成功是至关重要的。MDGSS的会前计划阶段一般按照四个基本步骤进行。第一,声明目的,定义问题。问题的定义必须反映出会议的目的。一旦这项工作完成了,就可以起草会议日程表。第二,确定产生想法、建立一致的途径。可采用的技术很多,具体采用哪一种由问题的定义和会议的目的来确定。在此基础上,可以描述群体和会议的详细协议,为会议的每一阶段分配时间。这时日程表可以完成了。第三,收集背景资料和相关数据。第四,准备展示。将材料按照日程表和会议进程组织起来。

当准备工作完成之后,就可以按照日程表进行会议了。对于参与者,会议本身在各个进程中最可见的部分。它也分四个步骤。第一,进行一次全体会议:传递会议目的,提供背景材料,对一致性要求作出解释,以供随后的分散会议使用。第二,想法的启发和捕捉。这方面所采用的技术和别的单群体决策系统(例如:脑风暴)非常相似。第三,想法的组织 and 评价。MGDSS中最困难的工作就是将各个群体产生的想法结合起来,并加以组织,以供进一步的深思熟虑。可采用分类和归并技术来解决这个问题。第四,想法的展示。用户可以对所展示的想法进行表决,也可将所展示的作为会议的最后结果。

会议中收集到的各种信息需要在会后作进一步的分析,分析的结果交给会议计划部门,再由计划部门分发给各参与者。

分类技术和归并技术对于MGDSS的实现非常重要。通过分类和归并,才能在看似杂乱无章的想法集合中建立次序和结构,按照各种各样的标准将它们分组。

各种想法都是用自然语言表达的。在分类之前,需要事先确定分类方案。一旦方案确定了,就可以提取出一些关键词,利用它们进行分类。比如,可以通过计算机检测某一关键词在哪些想法中出现了,将它们列出来,这样,我们就很清楚,那些想法是关于这一问题的。也可采用更复杂的技术,使得分类不仅是语义上的,还是句法上的。此外,还需要具有识别同义词的能力。这里就不作介绍了。

归并则是将同类的想法束集在一个中心主题周围。将想法按照相似性分组。可以采用的算法很多。一种基本的策略是检测各个单词在各想法中出现的次数,这样,每一想法都对应于一个向量。我们可以通过求任意两个向量的点乘或COSINE,来计算所对应的想法之间的相似度。

George Mason大学的AIRES(Advanced Integrated Requirements Engineering System)就是采用

了这种分类技术和归并技术,实现了对想法和概念的自动分析。

2 认识共享

分布式决策由大量的参与者作出。他们根据自己对形势的了解,在各自的范围内独立地工作;同时,他们又意识到他们之间存在着一定的联系。当参与者在工作中将这种联系考虑进去时,协调就变得非常必要了;这需要通过观点的提取来共享各自的观点,所以必须向参与者们提供帮助,使他们能描述自己对形势的理解。在此,我们关心的不仅是各种观点,更重要的是形成这种观点的背景材料,包括相关的知识、前提、假设,以及它们之间的因果关系。

可以采用认识图来描述观点及其影响因素。认识图作为一张有向图,其节点表示观点或因素,而弧表示源节点和目标节点之间的关系。认识图能够揭示个人的目的和逻辑,使人们能对以前所作的因果假设的合理性和有效性进行检测和评价,为人们提供了对形势深入了解、细致考察的机会。认识图还可以转化成矩阵形式,从而对决策逻辑进行定量分析。

和集中式决策中共享的上下文不同,在此,每一个观点及相关背景表达都有一个拥有者,只有拥有者有权对该表达进行修改。其它决策者们可以读这项表达,通过通讯网络将自己对该表达的疑问和建议传递给拥有者,但无权修改。

SPIDER 系统作为一种软件环境,支持观点提取和观点共享。在该系统中,一个表达由三部分组成:认识图,支持性的数据表格,文本注释;它们构成一个文件集。其核心是认识图。每个认识图都可以和其它的认识图、表格、注释或它们的组合相连接。同样地,表格可以和认识图等连接,注释也可以。

为了对一个复杂的文件集有一个总体的认识,SPIDER 提供了道路图。它由带有标题的图标组成,每个图标代表一个文件;这些图标由直线连接起来,每条直线代表一项文件连接。

对一个表达的分析可分为三个层次:全局的观点由道路图提供;在道路图中选定某个图标,则进入了文件层,用户可以同时浏览一个或多个文件;在一个认识图文件中,则可以观察各种细节内容。

对认识图的自动分析还有待实现,而表达方式和手段还需要进一步加强和改进。

3 三层网络结构

传统的计算机网络系统由客户机/服务器两层结构组成,用户接口等功能分配给客户机,而用户间相互作用的功能则由服务器完成。这种结构在多任

务情况下显得不足,因为用户间相互作用不仅存在于一个应用程序之内,还存在于各个应用程序之间。在设计这样的系统时,就需要把一个特殊应用程序内的协同功能和所有应用程序共享的协同功能区分开来。在这样的要求下,产生了三层结构:一个群件服务器,几个应用程序服务器,客户机。图 6 说明了这种结构。

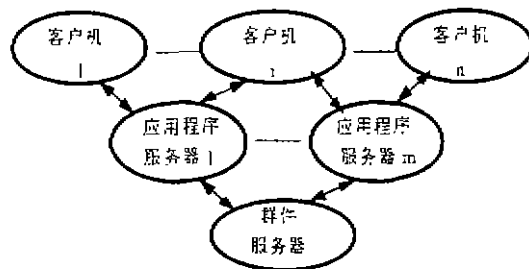


图 6 三层网络结构

群件服务器的基本目的是向所有群件应用程序提供必要的协同功能。比如,服务器记录着哪些应用程序正在运行,每一应用程序服务器及相关客户机的位置,以确保数据能正确传送。又比如,它管理着用户对应用程序的访问。当一个用户申请进入时,检查该用户是否有权进入。当一个登录的用户要求加入一正在进行的电子会议时,群件服务器检查该用户是否有权,如有,将用户的申请传给会议服务器(一个应用程序服务器),由会议服务器来决定是否接受该用户。应用程序服务器管理的是某一特殊应用程序执行时的协同及其它需要。例如,表决机制只适合电子会议使用,所以由会议服务器管理,显然要比群件服务器更适合。客户机则提供用户接口和一些应用工具。

DEEDS(Distributed Environment for Electronic Decision Support)模型说明了三层结构的可行性。它是在微软窗口环境下实现的,在通讯网络方面采用了 TCP/IP 协议。它包括了在线对话、群体绘图、电子会议三个应用程序,结构如图 7。但是,由于群件

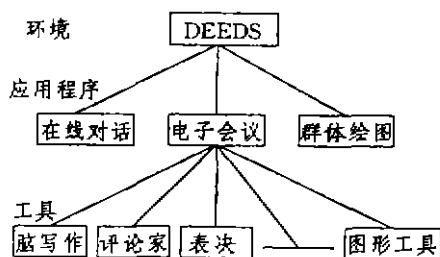


图 7 DEEDS 的系统结构

核心提供了公共的协同功能,新的群件应用程序能够被轻松地加入到该系统中去。

三、总结和展望

以上我们介绍了集中式群体决策系统和分布式决策系统。正如我们提到的,两者最大的差别在于前者的决策者们对面临的问题和环境有一个全面的认识,而后者则没有。由此引起了两者在模型和数据问题的处理上有很大的不同。

首先,集中式决策过程中产生的各种数据需要收集在一起,展示给全体决策者;而分布式决策的各个子单元只关心自己这部分的数据和对自己有影响的数据。其次,集中式决策中的数据,不论是匿名的还是记名的,在它产生之后,就不再保持和主人的拥有被拥有关系,即,群体中的任何一个成员都有权阅读和更改;而在分布式决策中,任何数据都有它的拥有者,只有拥有者有权改动,其它成员只能阅读,或通过对话向它的拥有者提出疑问和建议。第三,对待冲突问题的不同态度。集中式决策中的冲突一旦产生,必须通过对话、表达等方式来解决;而在分布式决策中,冲突产生的频率要相对高得多,可以通过对话来讨论,也可以任其存在。

所以,集中式决策中数据问题的重点是放在控制对数据的访问和实现数据共享上,而分布式决策则是致力于提供方法和手段,使决策者们更好地整理自己的数据、表达自己的观点,以供自己复审和他人观看。

集中式决策中的三阶段模型中,由于限制性的数据访问,艰难而耗时的合并工作,解决方案产生后不经讨论和评价即进行表决,而显得不那么尽如人意。共享上下文模型在数据产生的同时就将它并入公共的文件(共享的上下文)中。目前建议采用的方法是当一个用户在更新(加入、删除、修改)上下文时,就不允许其它用户同时作出更新;当用户数增加时,这一方法即显得低效了,这需要采用一种新的锁定模型,当用户对上下文中某一部分进行更新时,只对该部份锁定,其余部分仍然开放。

现有的集中式决策系统主要致力于如下主要问题:使群体通讯机制化;使决策进程结构化;结合各种支持决策的软件工具;引导群体的相互影响、相互作用。

此外,如何来描述会议过程,记录相关的数据;如何在会议中体现不同的地位(例如:采用特殊窗口),如何控制其作用和影响;对冲突问题的处理;会

议中对多种通讯方式的支持(例如:在线对话,电子邮件);对多种信息表达方式的支持(文本、图片、声音、录像)等等。这些方面已经有人做了不少工作,但是距离问题的解决,还需要进一步的实验和研究。

在分布式决策中,主要的问题有:从理论上研究分布式决策的策略、方法、决策过程模型;研究分布式系统的主要特征、分布式信息的表达、分布式决策信息的结构;研究分布式数据库、模型库、知识库的结构和管理;研究高效率、智能化的通讯网络结构和通讯方式;研究用现有的 DSS 装配 GDSS 的方法;完善 GDSS 概念,研究评价 GDSS 的指标体系和分析方法。

群体决策支持系统的研究和开发,不仅具有学术上的价值,还具有实用意义。在未来的社会中,各种组织的结构将越来越复杂,面临的问题也将越来越复杂。使复杂的组织解决复杂的问题,这正是群体决策支持系统的目的,也是 CSCW 的一个核心问题。

参考文献

- [1] Alessandra Agostini et al., A Prototype of an Integrated Coordination Support System, CSCW, 2, 1994
- [2] Randall Whitaker, GDSS Formative Fundamentals-An Interpretive Analysis, 同 [1]
- [3] Mark Pendergast et al., Alleviating Convergence Problems in Group Support Systems, CSCW, 3, 1995
- [4] Vitaly Dubrovsky et al., The Effects of a 'Distinct Window' Screen Design on Computer-Mediated Group Decision Making, CSCW, 4, 1996
- [5] Marshall Scott Poole et al., Conflict Management and Group Decision Support Systems, 1988 ACM, pp227-243, 1988
- [6] James D. Palmer et al., Multigroup Decision-Support Systems in CSCW, Computer, May 1994
- [7] Ting-Peng Liang et al., When Client/Server Isn't Enough: Coordinating Multiple Distributed Tasks, 同 [6]
- [8] 李海泉: 分布式决策支持系统, 西安石油学院
- [9] Jintae Lee, SIBYL, A Tool for Managing Group Decision Rationale, CSCW 90 Proc. October 1990
- [10] Richard J. Boland, et al., Sharing Perspectives in Distributed Decision Making, CSCW 92 Proc. November 1992
- [11] Michael J. Shaw, Mark S. Fox, Distributed Artificial Intelligence for Group Decision Support: Integration of Problem Solving, Coordination, and Learning, Decision Support System, 8, 1992