

78-80

群体决策支持系统与人工智能

沈锦涛

TP391

TP18

(杭州电子工业学院 杭州 310037)

A 摘要 群体决策支持系统(GDSS)正进入发展的第三阶段,与人工智能(AI)相结合。本文评述了当前 GDSS 与 AI 相结合的两个方面,一是用于知识获取,分析了它成功的原因并指出其特殊性,这是一条有可能突破专家系统开发“瓶颈”的途径;二是以分布式人工智能(DAI)为框架构成多主体(Agent)群体问题求解系统,分析了多主体 GDSS 与 DAI 的区别,着重论证前者较后者具有更高性能的原因。最后,提出 GDSS 今后可供发展的几个方面。

群体决策支持系统(GDSS),从八十年代初研制成功,在近十余年里,跳跃式地已经过了两个发展阶段,目前正处在第三个发展阶段。第一阶段,八十年代初—1989年。最初提出 GDSS 的概念,主要用于支持面对面的小群体作决策。这个时期理论上的奠基论文是 DeSanctis 和 Gallupe(简称 D 和 G)合写的“A Foundation for study of group decision support systems”^[1]。他们对 GDSS 的功能及发展提出了三个层次的概念。值得重视的是第三个层次,即在 GDSS 里引入人工智能,今天的发展表明他们对这个方向的指引是有预见性的,且是成功的,这个阶段开发的有代表性的系统是 Arizona 大学的 PLEXSYS, Minnesota 大学的 SAMM 及 Xerox PARC 的 Colab^[2]。第二阶段,1989—1992,这是 GDSS 飞速发展的时期。数量上在美国从有限的用户一下子发展到美国数据处理协会(DPMA)约有 1/3 的成员组织应用 GDSS^[3]。在功能和概念上由单纯地支持小群体决策发展到支持各种大群体的合作活动,并相应地开发出各种会议系统,有代表性的如 Arizona 大学的 EMS, New Yorks 州立大学的决策会议,美国陆军系统开发的 Video Conferencing System, 及加拿大 Toronto 大学的 Capture Lab。第三阶段,1993—,这是 GDSS 在应用和概念上的突破时期。其特征是和 AI 相结合,如 D 和 G 在 GDSS 的第三层次中所指出的,但范围更广泛。迄今的发展虽刚露端倪,却已呈现出很大的吸引力。本文将主要评述这一时期发展的两个方面:1. GDSS 用于知识获取;2. GDSS 与 DAI 的结合。

一、GDSS 与知识获取

自从 1965 年第一个专家系统 DENDRAL 开发

以来,知识获取一直是专家系统开发的“瓶颈”,在专家系统的开发过程中,几乎要占 2/3 的工作量。目前新开发的专家系统,知识往往要向多个领域专家去获取,知识工程师必须向每个专家反复询问,使矛盾更加突出。1993 年 Y. L. Lion 和 J. F. Numamaker, Jr(简称 L 和 N)巧妙地利用 GDSS 容许知识工程师可以同时平行地向每个专家获取知识,在解决“瓶颈”上取得惊人结果^[4]。他们为一个信息中心开发的一个咨询专家系统里,领域专家约有七人,另外还有些用户,仅一名知识工程师。整个过程分为三个阶段,知识获取规划、知识提取及知识验证,并分别作了三次现场询问。在 GDSS 支持下共召开了 12 次座谈会,累计时间仅为 6~7 小时,每次平均半个多小时。最后获得问题和解约 2000 条,经过原型系统验证及反馈又提出扩充建议 250 条。分析他们成功的经验有下列几点:

1. 直接在计算机上输入知识,使知识文件一步到位成为电子代码,省却了在口头询问中的抄录、整理和校对;

2. GDSS 支持每个领域专家可以平行、独立地输入知识;

3. 提供了一些结构化过程,如用于知识收集的懶格分析软件,对知识进行识别的论点分析软件,以及对问题进行规划、分类和验证的若干模型,使领域专家可以比较容易地回答各种询问,以及系统能自动对收集的知识进行分析;

4. 为专家们的交流提供了一种合作的气氛,这也是通过 GDSS 提供的一些技术来实现的,如电子头脑风暴法,Delphi 法及 NGT 等。

需要指出的是,L 和 N 这次试验的成功也有下

二个特殊原因:

1. 知识工程师对要开发的咨询专家系统中的知识比较熟悉,这不仅缩短了知识学习过程,而且可以比较明确地设计知识询问表以及易于识别和解决专家之间的矛盾知识;

2. 领域专家都是计算机专家,能熟练地操作和使用计算机。

在我国应用 GDSS 环境向领域专家获取知识时,可能还会遇到一个汉字输入问题,因为整个过程基本上在屏幕和键盘之间交互式进行,尤其在应用电子头脑风暴法时,领域专家要能随意地通过键盘来表达自己的意见可能会有一些困难,但是, L 和 N 的试验为打开知识获取这个“瓶颈”提供了很大的希望,可以继续研究和试验。

二、GDSS 与 DAI

如果将一个 DAI 系统里的若干个主体 (Agent) 看成是人员,并将其组织成一个 GDSS,这就构成了一个基于 DAI 框架的 GDSS。这个有趣的结构是由 Shaw 和 Fox 提出来的^[6],他们将这两种技术组合在一起,使 DAI 和 GDSS 的性能和应用范围都延伸到一个新的境地。

一个基于 DAI 框架的 GDSS 与单纯的 DAI 究竟有什么不同?本质上它们应是相同的,都属于 DAI,但结构形式上有区别,它们同样是由一群分散的、但相互有联系的主体所组成,目标是共同合作寻求问题的解,但前者组成了一个“集体”,而后者只是一个相互有联系的“集合”。正如人类社会一样,一个集体与一个集合在能力上往往会显示出差异,产生差异的原因主要有这样几点:①集体是有组织的,而集合是无组织的;②集体成员之间的联系更紧密,因此可采用多种或者一些特殊的协调方法;③集体成员之间通过相互学习能够增加求解问题的能力。下面分别说明。

1. 组织结构。主体的组织结构决定了被处理的信息量及主体之间的协调方式。在基于 DAI 的 GDSS 里,组织结构取决于求解问题的类型。一般有三种类型:①合作推理系统。这类系统的主体地位是平等的(也可以有层次,但最上层一定是平等的)。它们像一个小规模的 GDSS 一样,共同合作来求解问题。这类组织的一个特点是要有一个协调员进行协调。通常由一个主体承担协调任务;②分布式问题求解系统。这类系统一般将问题先分解成若干子问题赋给各主体,各子问题又可继续分解成子子问题赋给

子主体,然后由各子主体按各自目标求解最后进行合成。适宜于这类系统的典型结构是层次结构,合适的协调机制对这类系统特别重要;③连接体系统。这类系统中的主体只作为基本计算单元,所以严格说是非智能型的,但集合在一起可以很快解决某些复杂问题。文献中虽有这类系统的实例^[6],但它与 GDSS 毕竟有区别,本文将不作详细讨论。

2. 协调方法。对一个 DAI 来说,决定性能好坏的核心是协调。若协调不好,DAI 会弊多利少^[7],因此已研究出许多协调机制来适用于不同的求解问题。一般有这样一些:靠修改动作协调^[8];靠同步协调^[9];靠谈判协调^[10];靠机遇地满足目标协调^[11];靠交换偏好协调^[12]及靠约束推理协调^[13]。在基于 DAI 的 GDSS 里,这些协调机制应根据不同的求解问题综合应用。Shaw 和 Fox 还提出,既然基于 DAI 的 GDSS 也是一种 GDSS,一些结构化的协调过程,如 NGT、Delphi 及头脑风暴法等等也同样可以适用。这些协调方法目前已在试验之中。

3. 学习。无论是一个主体或一个 DAI 系统,学习始终是改进系统性能的一个重要过程。对一个 DAI 来说,改进性能是指:更好的协调及更有效的工作和资源配置,学习应包括两类:个体学习和群体学习。群体学习的方式为:信息共享、知识共享或主体之间更有效的通信,这意味着,一个合作得较好的群体在执行问题求解过程中由于共享信息和知识而仍在继续学习。这被称为 DAI 的额外智能,是一个智能化集体性能高于多个单主体性能之和的原因之一。

在单个主体里的概念学习是指对一组训练案例,先对一个案例提出假设,然后加入一个案例来修正假设,直至最后获得与全部案例相匹配的假设,即形成一个概念。在群体求解问题过程中,每个主体都承担收集数据、产生假设、评估假设和获得新概念的任务。一个主体产生的概念对另一主体来说即是新的事实,这就形成了一个群体诱导过程,它好比人类群体在求解问题中的相互诱导和自发激励,所以群体诱导过程也是使一个良好合作的 DAI 改进性能的原因。

一个基于 DAI 的 GDSS 由于它具有的组织结构、多样化的协调机制及群体学习功能,使其性能超过一般的 DAI 系统,同时由于它是由多个主体组成,求解问题实现全部自动化,这与仅作为工具支持群体成员决策的 GDSS 在功能上是一个飞跃。Shaw 和 Fox 曾研制了二个试验系统:一个用于市场、生产

及采购三方合作对策的 NEST;另一个是用于平行设计的分布式问题求解系统 Design Fusion,据称都得到比较满意的结果。

三、今后的发展方向

如上所述,在短短十余年里,GDSS 无论在概念、功能及用户数量上都有了飞速的发展。这种变化还将继续下去。对今后的发展方向,可以提出下列几个方面:

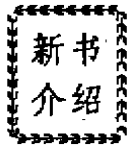
1. 随着通讯网络技术的迅速发展,特别是信息高速公路的建立,为远程网络的 GDSS 创造了一个良好的物理环境。如果再加上多媒体技术,使信息、图象和声音的传输会变得更快、更方便。开发这种环境下的 GDSS 会使群体成员虽相隔千里,却如近在咫尺,具有相当的吸引力;

2. 开发人机合作的问题求解系统。单纯的 GDSS 对问题求解只是一种支持工具,能协助群体成员分析策略、进行协调和作决策,但本身不能产生策略;而基于 DAI 的 GDSS 又完全抛开人员自行产

生策略、进行协调和作出决策。这显然像二个极端。人员主体与机器主体相比较,人更具有灵活性及丰富的知识,如果在基于 DAI 的 GDSS 环境里,人也能作为主体的一员参与,他同样能在机器主体产生新概念的启发下产生新的假设和概念,并可随机输入,而机器主体将这种新概念作为一个事实按概念学习机理又可产生新的假设,如此反复,人机共同合作对问题求解将会优于前述的两种极端;

3. 利用 GDSS 向多个领域专家同时获取知识仍是一个继续值得研究的方向。因为随着计算机技术的普及,各方面的领域专家会比较熟练地使用和操作计算机,而 GDSS 为他们提供了一个可以同时平行地输出知识和分析知识的环境,因此很有可能成为突破专家系统开发“瓶颈”的一条途径;

4. 对 DAI 的研究是当前计算机领域比较活跃的一个方向,DAI 性能的核心是协调,因此如何通过 GDSS 相结合,学习更多的协调机制以改进 DAI 性能仍是一个可继续研究的课题。(参考文献共 13 篇略)



并行分布式程序设计

《并行分布式程序设计》一书由华中理工大学计算机系刘健教授撰著,由华中理工大学出版社计划于 1995 年初出版。全书约 28 万字,共分七章,内容丰富、新颖,理论与实际并重。该书系统地研究分析了国内外近十年来在并行分布式系统软件方面所发表的大量文献,特别着重作者自己主持研制的 Fortran 并行化编译系统 HZPARA 和 HZPARA-I 的经验;在此基础上,进一步明确阐述并行分布式程序设计(特别是并行化编译)的数学模型、基本理论、基本方法及该学科的基本结构,该书主要面向高性能大型计算的并行分布式程序设计方法,重点讨论了并行划分、可执行条件、算法映射,进程调度以及并行程序设计环境与工具,并系统地论述了并行化编译系统实现中主要步骤的设计方法。

作者刘健教授长期从事分布式程序设计与并行处理软件的研究,他所领导的课题组于 1990 年研制成功了第一个 MIMD 系统上的并行化编译系统 HZPARA,较国外同类系统 Fortran D 编译系统早两年,而且 HZPARA 是全自动的, Fortran D 编译是半自动的。现在这个课题组又在研制并即将完成以并行化编译为核心的性能更好、功能更强的并行程序设计环境 HZPARA-I,并提出了有我国自己特色的并行化理论和方法。

《并行分布式程序设计》一书的出版无疑将丰富当前国内外计算机科学对并行处理软件技术(特别是并行化编译)的研究,促进并行分布式程序设计的推广应用,弥补该学科教材或专著的匮乏,引导青年计算机科学工作者把握正确研究方向,亦对该学科的专家、学者有较高参考价值。