

协同设计, Agent.

协同设计和多 Agent CAD 技术

殷勇 蔡希亮

(西安电子科技大学软件工程研究所 西安 710071)

TP391.72

摘要 The problem of how to construct a design system to support large routine design remains unsolved. Many approaches have been proposed, such as blackboard model, but they seem just suitable to small projects. Now, the agent approach shows us many features that make it a good candidate to solve the problem. In this paper, we briefly discuss the problem in a large design projects and give a detail discussion on the agent approach for cooperative design.

关键词 Collaborative design, Routine design, Agent, Multi-agents system

62=64 引言

当前的 CAD 技术正向着协同设计环境发展,是一个开放的、分布的、集成的协同工作环境。协同设计是一种系统化的方法,要求开发者从开始就考虑产品生命周期的全部因素,它的目标是缩短开发周期,改善产品质量,降低开发成本。协同设计环境是指支持相关人员集体进行设计工作的一种环境,包括计算机硬、软件技术、行为科学、认知科学等多学科支持的综合体。协同设计着重强调两方面因素,首先是设计人员采用群体工作的方式,以发挥各设计者的特长,其次是设计群体应有合理的结构。计算机支持的协同设计是以群体工作目标为核心,组织多学科人员进行协同,共同完成一个设计任务。目前,大多数设计支持系统仅支持单个设计者的设计,即使支持多用户的系统也很少考虑设计者之间多角色和多工作组的协同工作能力,因此必须研究支持协同设计活动的计算机辅助系统。

二、协同设计需解决的若干问题

设计任务按其性质可划分成创造性设计、常规设计和重新设计三类。常规设计是指能由少量的设计原语按确定的方法构造出最终设计对象的设计任务。设计结果表现为具有一定特征的已有部件的聚集体,部件通过具有特定属性的接口连接在一起,设计过程表现为一种逐步求精的过程。下面着重讨论协同的常规设计,若不加说明的话,协同设计即指协同的常规设计活动。

大型设计项目的一个显著特征是需要大量不同学科的知识才能完成整个设计活动。这一特性为协同设计活动带来了一系列的问题。

1)设计知识 大型项目的设计需要大量不同学科的知识。其中一部分为显式的知识,如手册、图纸、注释等,而另一部分却表现为隐式的知识,如设计经验、设计风格等。这一部分知识需在专家的帮助下,才能被发现、记录和维护。隐式的知识的形式化是非常困难的。

2)设计技能 设计技能是在特定的设计环境下运用显式和隐式知识进行设计的能力。如何在设计系统中反映设计技能是另一项非常困难的任务。

3)遗产问题 大的工程项目中,设计人员会使用各种不同的设计支持工具去完成设计任务。大型设计支持系统必需考虑如何将这类型不同的工具集成在一起。

4)人的作用 人是设计活动中重要的资源。由于专家知识的形式化是非常困难的,因而人成为任何设计系统中不可缺少的组成部分。设计系统中必需考虑人的作用。

5)元知识 元知识是关于设计的知识,包括各种技术注释,如各种设计中做出的决定和理由、设计错误等。这类知识极少在设计技术档案中体现出来,而这往往是设计项目成功的关键所在,这类知识的积累将导致专家的产生。另一类元知识是从设计过程中得到的经验和教训。大多数情况下,这些经验是不能被很好地记录下来的,这样,一旦设计队伍解散,这类元知识也就消失了。困难之处在于,很难找

到合适的方法和工具去抽象、记录和维护这类元知识。

6) 知识的一致性 许多研究人员认为设计在经过初始的一段时间后会由单一良定义的概念模型演化成最终设计,但在大的设计项目中,情况并不是这样的。事实是许多设计队伍同时进行设计,完成设计产品的不同方面要求,由于设计的复杂性,往往会产生设计的不一致。由于这一特性,很难实现单一的设计数据库。现实情况是设计人员独立进行工作,一段时间后通过协商,使矛盾的设计重新变得一致,因而大的设计工程表现为一种分歧/重新一致的设计过程。目前的设计工具缺乏这样的能力。

三、基于 Agent 的协同设计系统

在过去的若干年里,研究人员提出了各种方案来增强协同设计能力,其中最具有影响力的方法是基于黑板模型的构造方法,但并不令人满意。目前,随着 Agent 技术的发展,研究人员发现 Agent 表现出一些令人感兴趣的特点,十分有利于构造大型设计支持系统,下面我们将深入讨论这一问题。

3.1 黑板

迄今为止黑板模型是构造设计系统的主要方法。黑板模型中,不同的知识模块(知识源)可存取一个共同的数据库(黑板),每个知识源给出问题的部分解。

虽然黑板模型是一种流行的构造设计系统的方法,但我们认为这种方法存在许多问题:

- 黑板模型中由于公共数据库的存在表现出一种强耦合性。虽然数据的集中存放给数据的一致性带来了许多益处,但同时也成为系统的一个瓶颈,对于大的设计项目而言,这可能是致命的。黑板模型不具有可伸缩性。

- 存放于黑板的数据必须具有统一的格式,这样如何将已有的设计工具集成在黑板模型中成为一件十分困难的任务。

- 存储知识的时间和方式并不是很容易确定的,每个知识源都有自己的知识,知识可能表现为规则形式也可能是嵌入在代码中。

- 如何在黑板中存放元知识。

3.2 Agent

Agent 技术来源于分布式人工智能领域,现正向计算机领域的各个方面渗透。关于 Agent 的准确

定义,目前还没有一致的看法。大致上我们可以给出两种定义。

定义 1 (弱定义) Agent^[3]是具有下列特性的计算机软、硬件系统:1)自治性—可以不受人或任何外界因素的干涉而独立存在,对自己的行为和状态有一定的控制权;2)社会性—可以通过某种 Agent 通讯语言 ACL 和其它 Agent(包括人)进行信息交流;3)反应性—可以理解周围的环境,并对环境的变化做出实时的响应;4)能动性—可以主动地做出有目标的动作。

定义 2 (强定义) Agent 除了具备定义 1 中的所有特性外,还应具备一些人类才具有的特性,如知识、信念、义务、意向等精神上的观念和情感、能力等更抽象的概念。这一定义主要为人工智能领域所认可,引起的争论更多一些。

由于 Agent 呈现出许多有望解决建造大型设计系统困难的特点,目前利用 Agent 技术来构造设计系统成为一个研究热点^[3,4,5,7]。例如,FACT^[4]系统是一个典型的利用多 Agents 方案解决大型设计系统构造问题的例子。该系统由 Stanford 大学等单位共同完成,目的是为了如何支持涉及多学科设计人员、分布和多种设计工具的大型设计系统的构造问题。ACE^[5]是另一个用于电厂概念设计阶段的设计系统,它是基于演说模型(discourse model)的多 Agents 系统,具有发现设计冲突的能力,Agent 间可通过协商消除冲突,同样,ACE 也支持多学科设计人员共同进行设计工作。总的说来,构造方法是通过大量独立的模块的组合来构成系统。这些小的模块非常容易构造和维护,但困难之处在于 Agent 间信息如何进行交流和如何管理和组织大量的小模块。下面我们将详细讨论该方案的优、缺点。

- 由于 Agent 的自治性,系统的稳定性变得更好了,不会由于某个设计工具出错而导致系统瘫痪。

- 由于通过小的设计工具来构成整个设计系统,降低了系统复杂性。

- 运用 Agent 方法可以很方便地将原有的工具集成在一起,我们只需将它们封装在一起,对外呈现出一种具有良定义的 Agent 行为即可^[4]。

- Agent 具有本地知识,具有任务的局部表示,这意味着 Agent 可以运用任何所需的信息去完成工作,消除了黑板模型中数据格式必须一致的限制。关键之处在于 Agent 应具备由标准的 Agent 间信息交

换格式构造出内部表示的能力。

- Agent 系统中可以包含人的角色。人可以通过一个代理者 Agent 和设计系统相互作用,同时代理者可以和其它 Agent 相互作用,收集信息,帮助人完成工作。

- 某些 Agent 用来存放知识或记录设计决定,这样设计知识和元知识在系统中可以长期保存。原则上,这是非常容易实现的,因为所有的信息可以通过 Agent 间的通讯语言进行交流。

- 所有的 Agent 都可以异步地加入或离开一个工作系统。这对于大型设计系统是一个十分重要的问题。但在 Agent 加入系统时,如何建立系统的初始状态是一个困难的问题。

- Agent 结构表现出良好的伸缩性。

- 多 Agent 结构的实现十分困难,在这一领域鲜有十分成功的成果。

- Agent 间的通讯^[4]是另一个困难的问题。首先,Agent 间必须有一套能交换信息的最小协议集,其次,Agent 必须拥有共享的语义。目前的研究基本上都是根据言语学(Speech act)进行的。

- 大型系统中,完全的一致是不可能达到的,关键是如何通过协商来消除这种不一致。Agent 的协商能力为消除信息的不一致提供了可能性。

- Agent 系统最吸引人的地方在于它的认知方面的能力^[1]。Agent 可通过学习表现出一定的专家行为,成为设计助理。

结论 大型设计项目中,设计人员通常使用不同的工具,具有不同的知识,并不十分了解别人的行为,设计是许多设计人员行为的折中,如何构造能包含设计和产品知识的系统来改善设计行为和如何

将设计知识转交给设计产品的使用人员是构造设计系统的一个长期目标。传统的方法很难实现上述目标,基于 Agent 的设计系统似乎已给出了部分答案。在这一领域中,仍需做大量的工作去验证这一问题。

参考文献

- [1] J. Bates, The role of emotion in believable agents, CACM, 37(7)1994
- [2] C. Castelfranchi, Guarantees for autonomy in cognitive agent architecture, In: Intelligent agents, theories, architectures and languages, Springer-Verlag, 1995
- [3] M. R. Cuskosky et al., PACT: An experiment in integrating concurrent engineering systems, IEEE Computer, 26(1)1993
- [4] M. R. Genesereth and S. P. Ketchpel, Software agents, CACM, 37(7), 1994
- [5] J. Macguire et al., SHADE, Technology for knowledge-based collaborative engineering, J. of concurrent engineering, applications and research, 1(3)1993
- [6] Michal P. Case et al., Discourse model for collaborative design, Computer-Aided Design, 28(5) 1996
- [7] H. Park et al., An agent-based approach to concurrent cable harness design, AIEDAM, 8(1) 1994
- [8] M. Worldridge, N. R. Jennings, Intelligent agents, theory and practice, The knowledge engineering review, 10(2)1995

(上接第 89 页)

第二步,是采用“自顶向下”法:教师首先在自己熟悉的领域开设包含并行计算或面向对象专题的选修课程,然后再向不太熟悉的领域转移。

第三步,也就是本文所提出的最终目标,可采用“自底向上”的方法实现,即从导论性课程开始,到较高水平的课程,均把并行计算和面向对象的内容融入其中。这时,教师通过进修或讲授相关的选修课程,已积累了一定的经验。到时,包含合适专题的各种层次水平的教材相信已出版。本文提出的方法终

将被普遍采用。

结束语 并行计算及面向对象的概念应该贯穿计算机科学本科的整个课程体系中,以使 21 世纪的计算机研究开发人员能够灵活运用并行计算和面向对象技术去解决许多实际问题。作者认为,采用文中提出的一点一点逐步积累的方法,是让学生掌握不断出现的新概念、新方法、新技术的可行方法,因为这样不必去除传统论题的大量内容,同时也符合教学规律。(参考文献共 9 篇略)