

文章编号:1001-9081(2006)09-2182-02

基于多 Agent 技术的分布式协同设计结构的研究

冯相忠, 高 禹

(浙江海洋学院 信息学院, 浙江 舟山 316000)

(fxz@zjou.net.cn)

摘要:将多 Agent 技术引入协同设计中,使所构造的协同设计系统具有多 Agent 系统的分布性、协作性、智能性的特点。文中给出了 Agent 的结构、协同设计单元的多 Agent 组成和基于 Web 服务的整个协同设计系统的结构;并对协同设计系统实现的一些关键技术进行了讨论,包括 Agent 的创建、Agent 之间的交互、知识的共享和 Agent 之间的冲突的消解。

关键词:协同设计;多 Agent;Agent

中图分类号:TP18;TP393 **文献标识码:**A

Structure of collaborative design based on multi-agent

FENG Xiang-zhong, GAO Yu

(School of Information Technology, Zhejiang Ocean University, Zhoushan Zhejiang 316000, China)

Abstract: Multi-agent technology was introduced into collaborative design so as to make collaborative design system possess the advantages of multi-agent system, such as distribution, cooperation and intelligence. This paper presented the structure of agent, the composition of collaborative design unit: multi-agent and the structure of whole collaborative design system based on web server. And some key techniques for the implementation of collaborative design system were discussed, including creation of agent, communication and knowledge sharing between agents, conflict elimination of agents, etc.

Key words: collaborative design; multi-agent; agent

0 引言

多 Agent 技术在表达实际系统时,通过各智能体间的通信、合作、协调、调度、管理及控制来表达系统的结构、功能及行为特性。由于在同一个多 Agent 系统中各 Agent 可以异构,因此多 Agent 技术对于复杂系统具有无可比拟的表达力,它为各种实际系统提供了一种统一的模型,从而为各种实际系统的研究提供了一种统一的框架。

在协同设计中引入多 Agent 技术,将现有的软件系统、最终用户都可以用 Agent 进行封装。使所构造的分布式协同设计系统具有多 Agent 系统的特点。

1 协同设计系统的结构

1.1 协同设计中的 Agent 组成和结构

在协同设计中为了支持设计者的协同、设计工具的继承、设计者与 Agent 及 Agent 与 Web 应用的协作和交互,需要不同类型的 Agent 来完成。系统设计中主要使用了三种类型的 Agent,即应用 Agent、协作 Agent、设计者 Agent。

(1) 应用 Agent。主要用于各种分布式数据或应用资源的封装,包括工具封装 Agent、数据存储 Agent 等。

(2) 协作 Agent。用于完成各种协作处理工作,包括 Agent 协调器 (facilitator)、知识协作 Agent、冲突消解 Agent、群决策 Agent 等。

(3) 设计者 Agent。用于完成设计者与设计者、设计者与应用 Agent 和协作 Agent 之间的交互,包括界面 Agent、设计 Agent、过程配置 Agent 等。

三种类型的 Agent 在协同设计中分别面向不同的功能群体,在协同设计中起着不同的作用,它们分布在 Internet 连接

起来的协同设计动态联盟内,共同协作促进协同设计的设计效率。而每个 Agent 可以用如图 1 所示的通用结构来构建。

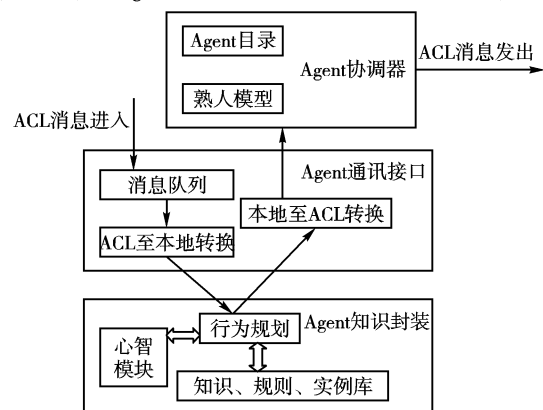


图 1 Agent 的通用结构

如图 1 所示,每个协同设计 Agent 中都包括 Agent 协调器、Agent 通信接口和 Agent 知识封装。Agent 协调器是 Agent 间相互通信的控制中心;Agent 通信接口为 Agent 与其他 Agent 的通信提供了一个转换接口;Agent 知识封装则包含了该 Agent 的心智模块和知识、规则、实例库等,以促进各设计 Agent 的智能决策,加快设计进程的协调。

1.2 协同设计单元的构建

在协同设计系统中,每个设计节点可以看作一个协同设计单元 (Cooperate Design Unit),即把每位协作设计者、设计 Agent 和相应的应用 Agent 组成一个 CDU,使 CDU 之间、协作的 Agent 间可以进行交互。但每个 CDU 的结构并不是完全一样的,即每个 CDU 完成的功能不同,CDU 的结构如图 2 所示。

图 2 中的各个 Agent 的功能是:人机交互 Agent 用于实现设计者和设计系统的交互;工具封装 Agent 用于实现把遗留的设计工具进行封装,使其集成到 CDU 中去;通信服务 Agent 为 CDU 的通信中枢,为各 Agent 之间提供了中间层服务,如名字服务、登录服务、注册服务等;节点通讯 Agent 用于和其他的 CDU 之间的交互;知识交互 Agent 用于实现基于本体的知识共享;过程管理 Agent 用于采用 workflow 技术定义和管理过程;冲突管理 Agent 用于协调各种冲突;决策协商 Agent 用于群体协商。

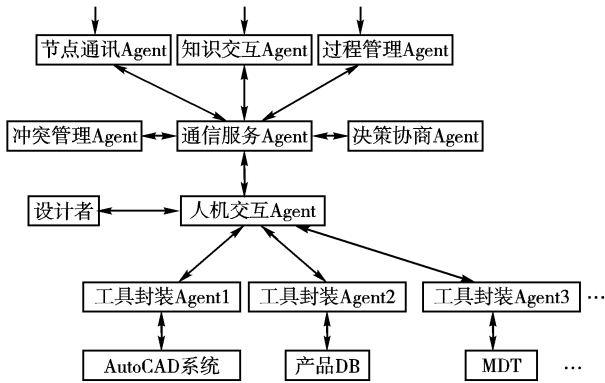


图 2 协同设计单元的组成

1.3 多个 CDU 组成的系统结构

一个协同设计系统可以由多个 CDU 组成,并且 CDU 之间是松耦合的、动态的组合方式,即设计者在设计中要完成一个设计任务,可以通过本地 CDU 动态地与其他多个 CDU 进行协商,分配任务,协作完成设计任务。结构如图 3 所示。

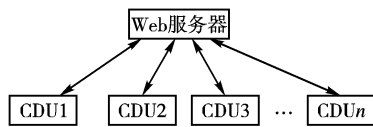


图 3 系统结构

2 系统结构实现的关键技术

2.1 Agent 的创建

基于多 Agent 的协同设计系统中,每个 Agent 的建立是实现系统的关键部分,本文中 Agent 的建立采用了 DECAF^[5] 平台。DECAF 是一个用于快速设计、开发、运行 Agent 的平台,它使 Agent 的创建过程变得相对简单。一个 Agent 的创建过程是:首先确定一个 Agent 要完成的任务,并把 Agent 的任务分解为子任务及子任务包含的相应动作集,在工具软件 Plan Editor 中绘制出任务实现的异或图,Plan Editor 工具软件自动生成 Agent 所需的规划文件;然后把规划文件作为 DECAF 平台的运行的参数,即可在 DECAF 平台上创建出一个 Agent,并可以在 DECAF 平台上运行所创建 Agent。

2.2 Agent 之间的通信

在基于多 Agent 的协同设计系统中,系统状态的更新和同步是通过系统内 Agent 之间的消息通信来实现的。Agent 之间采用 KQML(知识查询与操作语言)进行通信,KQML 是基于消息的 Agent 通信语言,如:

```
(ask-one
:sender &lt; agent1 &gt;
:receiver MatchMaker
:reply-with &lt; used&gt;
:ontology Matchmaker
:language DECAF
:content
(: keywords &lt; to match&gt;
:percentage &lt; number&gt;
```

)
)
ask-one 为通信原语,即一个 Agent (agent1) 询问另一个 Agent (MatchMaker),并且需要给出回答。这条消息的其他各部分表示消息的属性,其中“:sender”属性规定了消息的发送者;“:receiver”属性规定了消息预期接收者;“:reply-with”属性规定了消息的发送者是否期待回答;“:ontology”为本体属性,即定义消息使用的术语;“:language”属性规定了消息内容使用的语言格式;“:content”规定了消息的内容。

2.3 Agent 之间知识的共享

系统的构建中,采用了本体 (ontology) 技术进行知识的分类处理。本体的目标是捕捉相关领域的知识,提供该领域的共同理解,确定该领域内共同认可的词汇。从不同的层次的形式化模式给出词汇和词汇间相互关系的明确定义,通过概念之间的关系来描述概念的语义。

当多 Agent 协同工作时,Agent 之间必须对一些公共的术语和含义的理解达成一致。尽管公共基础只占 Agent 知识的一小部分,但它们对不同 Agent 之间的知识通信来说是非常必要的。而本体可以为不同 Agent 之间的知识共享提供语义基础。

2.4 Agent 之间冲突的消解

在协同设计中各个 Agent 之间的冲突是不可避免的,在系统的设计中主要采用如下一些方法来消解冲突:(1)通过过程调整消解过程冲突;(2)通过资源调整消解资源冲突;(3)通过统一术语和术语转换消除设计中的术语冲突。

2.5 系统中的知识决策

系统是通过不断的改进决策过程来推进产品数据信息流的,而产品信息流伴随协同设计的全过程。产品开发实质是一个逐步演化改进的过程。开发 Agent 和知识源之间的有序交互促进了产品信息流的不断向前流动。当某开发 Agent 系统需要进行决策时,向知识基决策系统的相应知识对象发送这一消息,启动决策过程,决策系统从产品信息库中获得相关信息,调用知识源的知识进行决策,并反馈信息。

决策系统的实现中利用元知识对知识进行分类、抽象和整理,动态地管理各知识对象,而对于遗留领域知识,用 Web 服务或 CORBA 对象封装为功能 Agent。

3 结语

在复杂产品设计中,通常涉及多学科的知识,需要引入多种设计方法和技术来达到设计目的,开发基于分布式的协同设计成为当务之急。本文在协同设计中引入多 Agent 技术构造了每个协同设计单元(CDU),使所构建的协同设计系统具有多 Agent 的分布性、动态性、协作性、智能性等特点。

参考文献:

- [1] SHEN WM. An experimental multi-agent environment for engineering design[J]. International journal of cooperative information system, 1996, 5(2):131-151.
- [2] CHEN SL, WU HZ, HAN XL. Collaborative Design Environment Based on Multi-Agent[A]. The 9th International Conference on Computer Supported Cooperative Work in Design Proceedings[C], 2004. 481-485.
- [3] 张洁,高亮,李培根. 多 Agent 技术在先进制造中的应用[M]. 北京:科学出版社,2004.
- [4] 闫俐,李继先,刘厚雪. 基于 Agent 的协同设计系统结构和实现方法[J]. 辽宁工程学院学报,2000, 20(4): 50-60.
- [5] GRAHAM JR, DECKER KS. DECAF-A Flexible Multi Agent System Architecture[J]. Autonomous Agents and Multi-Agent Systems, 2003, 7(1/2): 7-27.