

文章编号:1001-9081(2006)10-2480-03

基于 Agent 的协同设计中的 workflow 管理系统

陈 莉, 刘 弘, 李少辉

(山东师范大学 信息科学与工程学院, 山东 济南 250014)

(chli@163.com)

摘 要:针对 CAD 协同设计的特点, 结合状态图和 Agent 技术的优势, 提出了基于 Agent 的状态图 workflow 管理模式, 讨论了其关键技术和实现方式, 并以建筑设计为例进行了验证, 为协同设计中的 workflow 管理提供了有益的参考。

关键词:计算机支持的协同工作; CAD; Agent; workflow 管理; 状态图

中图分类号: TP311 **文献标识码:** A

Workflow management in Agent based collaborative design

CHEN Li, LIU Hong, LI Shao-hui

(School of Information Science & Engineering, Shandong Normal University, Jinan Shandong 250014, China)

Abstract: According to the characteristics of CAD collaboration design, combining the advantages of statechart and Agent technology, a workflow management model based on statechart and Agent was proposed. After discussing its key technology, its application in the workflow management of collaborative architecture design was demonstrated, and this provides helpful reference to the workflow management in collaboration design.

Key words: Computer Supported Cooperative Work (CSCW); CAD; Agent; workflow management; statechart

0 引言

随着 CAD 技术的推广和应用及 CSCW 的广泛应用, 怎样作好对协同 CAD 设计中的 workflow 管理也显得尤为重要。workflow 管理所解决的主要问题就是使多个参与者之间按照某种预定义的规则传递文档、信息或任务的过程自动进行, 从而实现某个预期的业务目标^[1]。人工管理过于简单, 因为通信不及时等原因会造成重复设计或错误设计, 不能够提高工作效率。

对于 workflow 管理系统, 文献[2]讨论了 Agent 在 workflow 管理系统中的应用, 但没有讨论 workflow 的表示模型; 文献[3]提出了基于 Agent 的 workflow 协调模型, 但其模型是基于面向对象的工作流元过程模型, 对于 CAD 相对简单的工作流过于复杂, 不利于用户使用 workflow 模型定义工作流。本文对于 CAD 的 workflow 管理, 根据其 CAD 协同设计自身的特点, 应用软件工程中的状态图方式来描述 workflow 图, 使创建的工作流规范化、简易化, 并且提出了用 Agent 来自管理协同设计中分布式环境下灵活的、动态的工作流的思想。

1 相关技术

1.1 状态图

状态图在模型化方式中构成一个描述状态和转换的可视化形式。它扩展了有限状态机的分类形式和状态转换图, 吸收了分层、并发、事件广播的概念, 以及相关的形式语法和语义, 这使得系统行为方面描述清晰化、严格化、人性化成为可能, 提供了模型形式验证和有效性的方法。这使得用户设置

的 workflow 很容易被系统识别, 且不会产生歧义。

1.2 Agent

Agent 是一种在异质的协同计算环境中能够持续完成自治的、面向目标行为的软件实体^[4]。它具有独立性、自主性和交互性。在信息流程中, Agent 通常充当一个中间处理部件, 它接收操作指令, 并依据自己的知识、规则的控制逻辑制定操作步骤, 完成要求功能^[1]。Agent 还具有很强的表达能力, 这意味着 Agent 之间可以通过公共的 Agent 通信语言进行通信, 独立于所在平台, 这解决了资源异构问题。在 workflow 管理中应用 Agent 可以实现无人管理, 并能及时的进行通信, 不仅提高的工作效率, 还提高了协同的有效性。

2 关键技术

2.1 基于状态图的工作流图 (WFCBS)

基于状态图的工作流图^[5]采用了 Harel 的状态图^[6]的基本语义, 并对相关工作流部分进行了相应的扩充。基于 Harel 的状态图, 我们将工作流图定义为 11 元组: $WFCBS = \langle W, E, r, \rho, \Psi, \delta, C_p, E_p, A, R, T, \rangle$, 其中: W 表示了 workflow 中任务集合; E 表示原子事件的集合; $r \in W$ 表示初始状态即初始任务; 层次函数 $\rho: W \rightarrow 2^W$ 定义了每一个状态的子状态集合; $\Psi: W \rightarrow \{AND, OR\}$ 定义了每个复合状态的合成类型; $\delta: W \rightarrow 2^W$ 是缺省函数, 定义了一个状态 W 的一系列子状态集作为 W 的缺省集, $\delta(W)$ 描述 S 被激活后进入的子状态; C_p 是基本条件集, C 是条件集, 包括 $T(\text{true}), F(\text{false})$; E_p 是基本事件集, E 是事件表达式集; A 是一个动作集, 满足某条件所进行的动作; $R \subset E \times A$ 是由一系列序对 $(e, a) \mid e \in E, a \in A$ 生成的

收稿日期: 2006-04-07; 修订日期: 2006-07-14

基金项目: 国家自然科学基金资助项目 (60374054); 山东省中青年科学家奖励基金项目 (304065); 山东省自然科学基金资助项目 (Y2003G01); 山东省科技攻关制造业信息化项目 (021050101-1-7)

作者简介: 陈莉 (1974-), 女, 山东青岛人, 讲师, 硕士, 主要研究方向: 软构件、建筑协同设计环境、图形学; 刘弘 (1956-), 女, 山东济南人, 教授, 博士, 主要研究方向: 软件智能化技术、软件开发环境; 李少辉 (1978-), 男, 山东烟台人, 硕士研究生, 主要研究方向: 协同设计。

标签集; $T \in 2^W \times R \times 2^W \cup H$ 是转换集。

可并行的任务的工作流如图 1 所示。

workflow 中每一个任务都是一个状态,每一个状态又分为若干子状态,包括:工作、等待、停止三个子状态,每个子状态都有其转换条件,并产生相应的事件。我们将 workflow 中的任务分为三种类型:初始任务、中

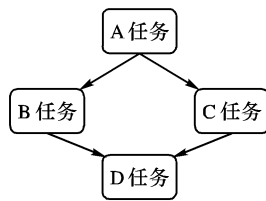


图 1 可并行运行的任务

间任务和终止任务。图 2 为中间任务的状态图表示,其中 S0 为工作,S1 为停止,S2 为等待子状态。

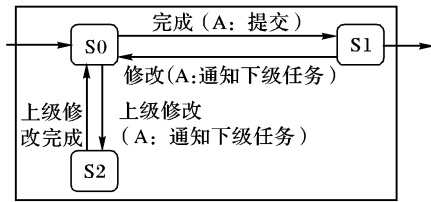


图 2 中间任务的状态图表示

2.2 Agent 的设计

workflow 管理中的 Agent 可以表示为如下三元组 $\langle DS, AS, RS \rangle$ 组成,DS 是 Agent 的数据集合,记录 Agent 中所有定义的数据;AS 是 Agent 的活动集合,记录 Agent 可完成的各项活动;RS 是 Agent 的规则集合,记录 Agent 进行各项活动的各种规则。由于 workflow 管理中的 Agent 任务比较单一,相关的 Agent 也较少,所以在 Agent 通信中采用点对点的通信方式,增强了保密性和安全性。在实现中我们将 Agent 分为两种:客户端 Agent 与服务器端 Agent。

2.2.1 客户端 Agent 的实现

客户端 Agent 负责监测设计人员的工作状态并及时传送给服务器端 Agent。客户端 Agent 主要是负责设计人员这一方的管理工作,如通知修改,向服务器 Agent 发送修改请求或提交设计人员完成的资料等。具体算法如下:

1) 如果接到服务器 Agent 的允许工作的命令,接受服务器 Agent 传来的有关图样和设计文档,并通知设计人员可以开始工作。记录设计人员该任务的状态为工作 S0 提交给 workflow 管理系统。

2) 如果接到服务器 Agent 发送的上一级任务进入修改过程中,须停止工作,则通知设计人员停止工作并保存设计人员的有关工作文档。记录设计人员该任务的状态为等待 S2 提交给 workflow 管理系统。

3) 如果接到服务器 Agent 发送的继续工作的通知,接受服务器 Agent 传来的新资料,恢复上次中止的工作状态,并通知设计人员继续工作,将新的图样及文档交给设计人员。记录设计人员该任务的状态为工作 S0 提交给 workflow 管理系统。

4) 如果接到服务器 Agent 发送过来的要求修改的请求的通知,将请求者和请求原因交由设计人员。

5) 如果本端设计人员提出的修改请求,若这个请求修改是关于本任务的,则通知服务器 Agent;若这个请求是修改本任务前几级的(即状态图中该任务的祖先),则向服务器 Agent 发出请求。

6) 当设计人员完成工作或修改完成时,由客户端 Agent 将其资料上传,通知服务器端 Agent。记录设计人员该任务的状态为停止 S1 提交给 workflow 管理系统。

2.2.2 服务器端 Agent 的实现

服务器端 Agent 负责整个 workflow 的管理与协调。接受客

户端 Agent 发送的各种信息,并根据具体信息,依据 workflow 图的表示自主进行下一步工作的安排,将各种信息返回给客户端 Agent。具体算法如下:

1) 根据 workflow 图中的初始状态(初始任务)安排给相应的客户端 Agent。

2) 接受客户端 Agent 的完成显示,接受客户端提交的资料,存档。根据 workflow 图中的顺序将相关内容传给下一个任务的有关客户端 Agent,通知其可以开始工作。

3) 接受客户端 Agent 对自己任务的修改请求,依据 workflow 图通知其后面相关的各个任务的客户端 Agent 停止工作。如图 1 中,若 C 进行修改,则需通知 D,而不需要通知所有参加协同设计的人员,其他设计人员仍可继续进行工作。若 D 还有下一级则也需要通知其停止任务。

4) 接受客户端 Agent 对上级任务的修改的请求,并将该请求内容和请求者发送给相应的客户端 Agent。

5) 接受客户端 Agent 修改完成的通知,接受其上传的新的资料,并通知 workflow 图中其下一级任务的客户端 Agent 继续工作且将新的资料传过去。

2.3 workflow 图与 Agent 的结合

在某一 workflow 中,用户 1(US1)与用户 2(US2)所接受任务的关系在 workflow 图中是相连的关系,用户 1 完成后用户 2 才可工作,图 3 显示了用户任务的状态及 Agent 在此状态下所作相应反应,大多 Agent 具体工作并未在此图中体现。

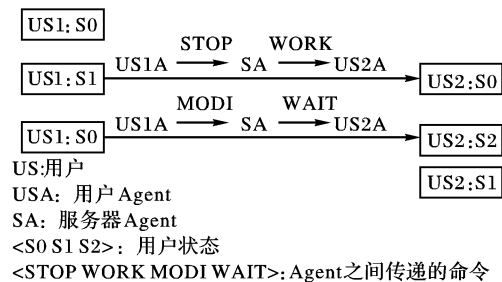


图 3 workflow 图与 Agent 的结合示例

3 实现

workflow 的管理系统分为客户端和服务器端两部分,主要部分集中于服务器端。总体设计如图 4 所示。

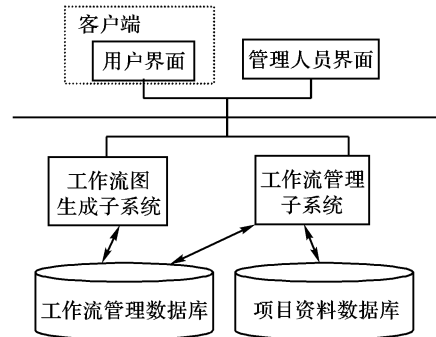


图 4 workflow 管理系统总体设计

在服务器端,管理系统分为两部分:

1) workflow 图生成子系统:本系统主要任务是根据项目要求生成 workflow 图,大部分的设计单位都是对某一方面进行专门设计,所以 workflow 比较固定,有差别也不是很大。管理人员可以按照行业特点作几个固定的 workflow 模板,供大多数项目来用。对于某一具体项目按照 workflow 图中的表示,根据各设计人员的具体情况(有几个项目处在工作状态,有几个

项目处在等待状态),安排有关设计人员进行工作,这样就组成了一个包含几个设计人员临时小组负责共同完成这一项目。将项目信息(如项目号)、设计人员信息(如终端地址)及项目的工作流程图交由服务器 Agent,由服务器 Agent 按照流程图来统一管理该项目,直至项目结束。由于每个设计人员有可能同时参与多个项目的设计,所以每一项目的安排信息都需要详细记录存档,以免发生传送、管理的错误。这些数据都存于 workflow 管理数据库中。

2) 工作流管理系统:本系统接受生成子系统的任务开始进行有效的管理,记录每个项目中各个设计人员在设计过程中的设计状态,如:修改次数、修改原因、停止次数、停止原因、当前状态等等,直至项目完成,并将其提交给工作流管理系统,由管理人员作为安排其他任务的依据。大部分任务由服务器 Agent 自主完成。这些信息存于工作流管理数据库中。对于某设计人员完成的资料存入项目资料数据库,若进行过修改将版本信息一并存入项目资料数据库中。

4 具体应用

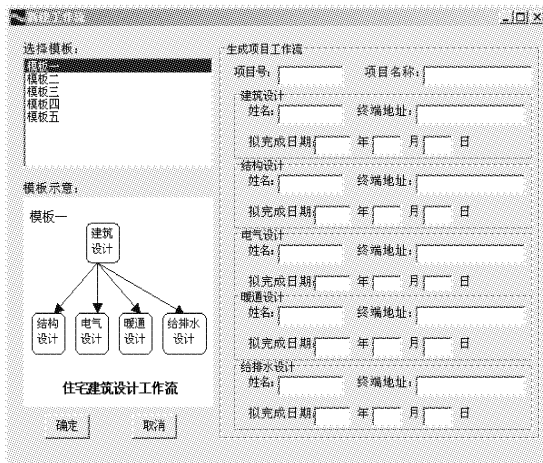


图5 新建项目工作流界面

已将本设计思想应用于建筑设计的协同设计方面,对于民用建筑设计过程相对简单,管理起来更加方便快捷。假设在一住宅楼的设计中,所需的工作流程如图5。

(上接第2472页)

作在整个文档开发中,甚至整个软件开发中,需要统筹安排。

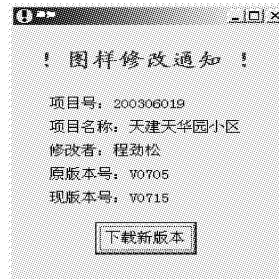
为了方便文档测试的组织工作,文档测试的组织方式有:

- 1) 单元测试:将 UD 分拆成小的单元,单元可以按文档的节或章划分,以单元为单位组织测试。
- 2) 分类测试:根据文档的组成,将版面、功能叙述、示例、图、表等,分类进行测试。
- 3) 整体测试:将所有 UD 统一进行测试,测试不同文档、文档的不同部分之间的风格、内容、术语、引用的一致性、可追溯性。
- 4) 可理解性测试:测试文档描述是否清晰,易于读者的理解,特别是不能出现概念错误。
- 5) 易用性测试:测试编排方式和描述方式便于用户使用,如模拟用户查找文档,测试在文档中定位信息的方便程序。

5 结语

文档是用户正确使用程序的基本保证,是软件开发的成果之一,UD 的质量与程序一样也需要得到保证。文档与程序特性不同,决定文档测试技术、组织方法与程序不同。经过长期的研究和实践,提出了我们对文档测试的研究成果。后

在本设计中除了建筑设计需要先进行,其他设计任务可同时进行。建好之后,Agent 便可以根据它的任务及设计人员的工作状态进行一系列管理。本工作流应该是最简单的工作流程了,但建筑设计中有其特有的特点。如在建筑设计中,并不是所有设计的修改都在图样上显示出来,有些不是很重要,不需要作大修改的地方,只要写一个变更文档就可以,对于这个可以不作为流程图中的修改处理,即不需要下一级任务停止工作等待,只要上一级任务完成变更后通知下一级任务将变更文档传送就可。图6是某上一级任务设计人员进行修改后,将新版本通知下一级任务的设计人员界面。



5 结语

本文讨论了协同设计中的工作流管理问题,针对 CAD 协同设计的特点,提出了基于状态图的工作流模型,提出的基于 Agent 的管理系统,描述了其系统结构,并在最后提出

图6 通知下一级设计人员界面了在建筑协同设计中的应用及相关问题。应用于某建筑设计院,减少了人工管理带来的许多弊端,有效的提高了工作效率。

参考文献:

- [1] 史美林,向勇,杨光信. 计算机支持的协同工作理论与应用[M]. 北京: 电子工业出版社, 2000.
- [2] 李红臣,史美林. Agent 在工作流管理系统中的应用研究[J]. 通信学报, 1999, 20(9): 16-22.
- [3] 朱云龙,薛劲松,李红信,等. 基于 Agent 的工作流协调模型研究[J]. 小型微型计算机系统, 2000, 21(7): 737-739.
- [4] 刘广钟,曾聪文. 基于 Agent 的分布式计算的研究[J]. 计算机工程与应用, 2002, 21: 88-89.
- [5] 陈莉,刘弘. 协同设计中基于状态图的工作流表示模型[J]. 计算机应用研究, 2004. 6: 18-19.
- [6] CRISTINA M, DE OLIVEIRA F. A StareChart-Based Model for Modeling Hypermedia Applications [DB/OL]. <http://www.dct.ufms.br/~turine/hmbs/ACM.pdf>.

续还将开展测试技术的深入研究,开发文档测试的支持工具。

参考文献:

- [1] PATTON R. Software Testing 1st Edition[M]. Sams Publishing, November 2000.
- [2] BROWN A, PATTERSON DA. To Err is Human. Proceedings of the First Workshop on Evaluating and Architecting System dependability (EASY'01)[M]. Göteborg, Sweden, July 2001.
- [3] PRESSMAN RS. SOFTWARE ENGINEERING: A Practitioner's Approach Sixth Edition[M]. McGraw Hill, 2005.
- [4] BEVAN N. Tutorial 11 — Industry Standard Usability Tests, Interact'99 Conference[Z]. Edinburgh, 1999.
- [5] ISO 9241-11: 1998, Ergonomic requirements for office work with visual display terminals (VDTs) —Part 11: Guidance on usability [EB/OL].