

文章编号: 1001-2486 (2000) 01-0011-04

一个分布式并行设计集成框架系统的研究与开发^{*}

李思昆, 郭 阳, 彭宇行, 杨 军

(国防科技大学计算机学院, 湖南 长沙 410073)

摘 要: 复杂产品的设计日益复杂, 推向市场的时间日益缩短, 并行设计技术得到广泛应用。文中实现了一个支持复杂产品多领域并行设计的集成框架 YG-bridge。该集成框架的开发采用符合 CORBA 规范的 Client/Server 中间件, 实现了基于设计体模型的设计数据管理、基于语义模型的工具封装和基于扩展网络图模型的设计过程管理。YG-bridge 已在实际设计工作中得到较好的应用。

关键词: 电子 CAD; 并行设计; 集成框架; 分布对象; Petri 网

中图分类号: TP391.72 **文献标识码:** A

Research and Implementation of a Distributed Framework for Concurrent Design

LI Si-kun, GUO Yang, PENG Yu-xing, Yang Jun

(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: As the design of products is becoming more complex, and the time it takes for these products to go on the market becomes shorter, the concurrent design technique is widely used. This paper proposes a distributed framework—YG-bridge, which supports the concurrent design of multi-domain complex products. Based on Client/Server middleware, the framework realizes design data management based on design body model, tool encapsulation based on semantic model, design process management based on the extended net graph model. YG-bridge has been applied successfully in practical design work.

Key words: electronic CAD; concurrent design; framework, distributed object; Petri net

并行设计采用集成化和并行化的思想设计产品, 在产品开发的早期就充分考虑产品生命周期中的所有因素, 以达到提高质量, 缩短产品开发周期的目的。集成框架是实现并行设计的基础支撑环境, 它是在异构、分布式计算机环境中实现企业产品全生命周期的信息集成、功能集成和过程集成的软件系统^[1]。现有的集成框架系统存在以下问题: 缺乏形式化的过程管理和数据管理的理论; 在总体结构设计和部件设计中未考虑各种设计方案和设计途径的管理; 设计数据一致性检验能力不强等^[2, 3]。

1 YG-bridge 集成框架的主要功能与特点

针对并行设计对集成框架的迫切需求, 结合大型电子设备的研制背景, 我们成功开发出分布式集成框架系统——YG-bridge^[4], 该系统是国内最早的基于 CORBA 规范和软构件技术的分布式集成框架之一。它不仅能实现分布环境中产品数据的统一管理, 还能够很好地支持并行设计。

YG-bridge 的主要功能及特点如下:

(1) 提供了直观友好、易于操作的图形用户接口, 方便最终用户快速设计电子系统与设备, 方便工具开发者开发和集成新的 CAD 软件。

(2) 实现了 CAD 应用软件和系统的封装与启用。封装工具允许封装后的 CAD 软件方便地插入到集成框架中, 并由用户按需要启动 CAD 软件。

(3) 具有产品数据管理功能, 包括: 建立产品的结构描述, 实现产品数据的配置管理; 实现产品

^{*} 收稿日期: 1999-06-03

基金项目: 国家 863 计划项目资助 (863-511-04-01)

作者简介: 李思昆 (1941), 男, 教授, 博士生导师。

数据版本管理, 包括数据提交、审批和发放; 支持产品设计过程定义, 实现设计过程管理; 支持项目组角色定义, 实现权限管理及设计进度管理。

(4) 支持产品的并行设计

- 系统以 Client/Server 计算模式为基础, 支持异构计算环境, 用户可在异地、异种工作平台上并行、协同地进行产品设计;
- 对应用工具的封装, 便于有效管理应用工具产生的信息, 便于应用系统之间的信息传递与交换;
- 为产品建立了统一的设计体模型, 以该模型为基础, 设计者可并行地对产品的各部件展开设计, 部件间的信息交换、一致性控制及管理由系统统一实施;
- 项目管理中的团队管理功能可有效地协调设计成员之间的工作;
- 进度管理和过程管理可有效地对并行设计过程进行规划、控制和监测, 为并行设计中的过程集成提供必要的支持;
- 域对象封装管理构件和虚拟原型数据管理构件, 有利于在上游设计中考虑下游设计的各种因素, 最大限度地实现并行设计。

2 关键技术

2.1 采用符合 CORBA 标准的 Client/Server 中间件

分布对象技术始于 90 年代初, 目前已成为异构环境下建立应用系统集成框架和标准构件的核心技术。YG-bridge 以网络环境下的分布对象技术为支撑, 以 Client/Server 方式工作。共享数据存放在服务器上, 各用户与客户机上的集成框架用户界面交互, 通过软构件提供的服务对数据进行访问。软构件还可以对用户的设计活动进行协调, 确保用户工作在一个高效、一致而安全的环境中。YG-bridge 基于 StarBus 中间件来开发, StarBus 是我系自行开发的符合 CORBA 规范的 Client/Server 中间件。利用 StarBus 开发分布式集成框架能够有效控制集成框架软件开发的复杂性, 无需考虑如何管理和访问异地对象, 集中精力实现应用对象, 从而大大减少集成框架开发的工作量。同时将 StarBus 作为“软总线”, 易于设计可重用性、移植性和互操作性好的对象构件, 提高集成框架系统开发的效率。

2.2 基于设计体模型的设计数据管理

YG-bridge 中的数据以设计体模型来组织。设计体模型能清楚地描述设计对象间的层次分解关系及属性版本关系, 它将与设计对象相关的设计对象分为设计主体、设计实体及设计实例, 构成设计主体到元设计实体的多层结构。每个设计实体的数据按属性类存放, 属性类之间可以有层次关系, 只有位于叶节点上的属性类才能指向实际设计数据。基于设计体模型, 从两个方面实现设计数据管理功能:

(1) 多数据库管理: 采用松耦合结构, 对设计过程中各个阶段的设计信息实施集中统筹管理, 为设计人员提供高效、合理、可靠的设计环境。服务器数据库作为中心数据库存储全局数据。各 CAD 工具的数据库存储局部数据, 通常是针对某个设计阶段的, 其组织和管理保持相对稳定和独立, 相互之间基于一定的转换策略集成。这种松耦合方式, 既可以使用户对数据的访问通过中心数据库统一实现, 又保证了中心数据库与子数据库之间的分离结构, 从而使各部分保持相对的独立性。

(2) 高层数据管理: 包括版本控制和配置管理、分布式访问与并发控制、数据安全性保证等。版本管理基于层次网络模型实现, 能有效地满足用户保存和读取所需要的历史信息的要求; 配置管理支持复杂对象的存储与管理; 分布式访问与并发控制保证了数据的一致性。

2.3 基于语义模型的工具封装

工具集成和工具封装机制是实现工具的互操作, 及工具间的数据集成、控制集成和过程集成的重要手段。工具封装包含两个方面内容: 一是为工具提供注册、激活和管理的标准接口, 使得框架能以正确的语法、参数和数据调用工具; 二是建立框架数据与工具数据之间的关系, 通过接口连接, 实现框架与 CAD 工具间的双向操作。前者称为工具的功能封装, 后者称为工具的数据封装。

我们提出的工具封装方法的基本思路包含以下几个方面:

(1) 基于 Client/Server 平台, 将工具封装界面与封装服务相分离, 通过 CORBA 中间件连接界面与

服务, 这样既可解决工具间的异构性, 又保证了系统具有很好的开放性和可扩展性;

(2) 将工具封装分为功能封装和数据封装两级, 所有设计工具必须进行功能封装, 对采用标准硬件描述语言的工具, 在功能封装后还要进行数据封装;

(3) 以 CAD 数据文件的内部语义为基础进行数据封装, 细化数据管理粒度。设计结构信息以设计对象的粒度管理; 设计表示信息以文件方式在工具间传输。

2.4 基于扩展网络图的设计过程管理

为满足并行设计过程管理的需要, YG-bridge 采用基于扩展网络图模型的设计过程管理方法, 支持项目级与任务级两种层次的过程定义, 以网络图为基础实现项目管理与规划, 以 Petri 网为基础实现任务流管理和数据一致性管理, 较好地支持集成化的项目、过程和数据管理。

定义 1 网络图是由节点和边构成的有向图, 可用元组 $GN = (V, E, A)$ 表示。V 是节点的集合, 节点表示一个或若干个设计项目的开始; E 是边的集合, 边表示子设计项目, 子设计项目的实施往往可分为多个阶段, 实施过程可用任务流图 GT 表示; A 称为 E 的属性。

定义 2 子设计项目的任务流 $GT = (T, S, E)$ 描述了子项目设计过程中的信息流, T 为任务节点的集合, 任务节点由任务描述符来标识, 可以是终结或非终结的, 终结任务节点表示 CAD 工具或其它可执行程序, 非终结任务节点表示抽象的任务, 可以用不同的工具或工具组合完成; S 为描述节点的集合, 描述节点以描述类型来标识; E 为边的集合, 连接描述和任务。

定义 3 项目的网络图及各子项目的任务流图构成项目的扩展网络图。

定义 4 任务流 GT 可用 Petri 网图 $EN = (P, T, F, M)$ 描述, 其中, $P = \{p_1, p_2, \dots, p_i, \dots, p_n\}$ 是位置的集合, p_i 代表数据文件; $T = \{t_1, t_2, \dots, t_i, \dots, t_m\}$ 是转换的集合, t_i 代表任务步或控制步; $F = (P \times T) \cup (T \times P)$ 表示 EN 的流关系; $M_s \square P \rightarrow I, I = \{0, 1\}$, 是标记函数, 表示数据文件的状态。M = $\{M_s$ 是位置 p 的标记 $p \in P\}$, M_s 称为格局。

(1) 基于网络图的项目规划

项目规划在项目实际开始前, 描述设计项目, 估计完成项目和各子项目所需的时间, 产生网络图, 调度子项目并分配资源, 监控整个项目的进展状况。项目规划包括时间规划与资源规划两个方面。时间规划的基本思想就是找出关键路径, 优化安排关键设计上的人力、物力, 压缩关键设计的作业时间, 有效地调整项目进度。时间规划结束后, 需要考虑资源因素, 如果在子项目进行的时间内资源得不到满足, 则按以下步骤处理: ① 询问设计环境模型, 所需资源何时能满足; ② 在网络图中插入一个资源分配任务; ③ 在资源分配任务与使用资源的任务间插入依赖关系; ④ 产生新的项目网络图; ⑤ 计算新的时间规划, 找出新的关键路径。

(2) 基于 Petri 网的任务流管理

任务流的构造: 通过界面提供的任务流编辑构造工具完成对任务流的定义。

任务流的显示: 显示各任务步定义的内容 (任务步的次序关系、工具名、任务名等) 以及项目实施过程中各任务步的执行状态 (就绪, 执行, 完成等)。

任务流图到 Petri 网图的转换: 用户输入任务流图, 并完成任务流实例化后, 系统自动按转换规则将任务流图转换成 Petri 网图作为内部的表示。

任务流执行控制: 包括如下功能:

- 应用系统的启动: 根据 Petri 网的点燃规则控制任务流执行及应用系统的启动;
- 串并行控制: 通过转换步与控制步的激活和点燃机制实现任务流程的串行和并行控制;
- 控制步可实施设计回溯控制;
- 设计任务的同步控制: Petri 网执行规则可以确保设计任务按一定次序无冲突地运行;
- 设计历史记录: Petri 网的点燃序列 $(t_{j(0)}, t_{j(1)}, t_{j(2)}, \dots)$ 可以提供任务流执行情况的记录。

(3) 数据一致性管理

对数据文件 $p_i, p \in P$, 转换 $t \in T$, 若 $p_i \in t$ 且 $p \in t, i = 1, 2, \dots, n$, 根据 p 的数据版本号, 判断 p_i 是不是对应的版本号。此过程一直进行下去, 直至达到扇入数为 0 的节点。若所搜索的数据均

为最高版本, 则最高版本的设计数据是一致的; 否则, 可以定位导致数据不一致现象的设计阶段。

3 系统结构与开发方法

3.1 系统组成结构

YG-bridge 集成框架采用“框架总线+ 软构件”的结构, 如图 1 所示。

(1) 集成框架软总线

集成框架软总线的主要功能是:

① 实现软构件到集成框架的封装, 对软构件实施统一的和一致的管理, 方便软构件间的通讯和数据交换。

④ 实现并行设计集成框架用户界面与各软构件的连接, 提供一组基于 IDL 的一致 API, 描述各构件的特性、行为和接口。

(2) 软构件

软构件是定制式的, 针对特定应用领域 (如电子产品设计), 是可重用的软件模块, 不作为独立的产品。YG-bridge 集成框架实现了六个软构件: 版本管理构件支持对设计数据的版本管理, 维护版本一致性; 配置管理构件实现对产品结构及数据的组织与管理; 项目管理构件对设计进度进行规划和跟踪, 并支持对设计团队的管理; 设计过程管理构件以任务流为基础, 支持对设计过程的建模、控制、跟踪与监测; 虚拟原型数据管理构件以配置管理为基础, 管理与虚拟原型相关的数据; 域对象管理构件实现域对象的封装与管理。

3.2 YG-bridge 集成框架的开发方法

YG-bridge 集成框架的开发是一个渐进的过程, 软构件和用户界面的开发相对独立, 用户可根据需要, 不断增强系统的功能, 这就保证了系统功能的开放性和开发过程的渐进性。

(1) 软构件开发

① 分析用户需求, 确立各软构件的主要功能;

④ 建立各软构件的对象模型;

④ 使用 IDL 定义软构件的接口;

¼ 利用 StarBus 的 IDL 编译器, 生成软构件的实现模板及其在客户方的代理;

½ 利用 Visual C++ 实现各软构件具体的服务功能;

¾ 编译、连接、调试应用程序。

(2) 用户界面的实现

在微机和工作站上, 分别用 Visual C++ 和 Borland Jbuilder 开发集成框架用户界面, 两套界面风格一致, 并采用同样的接口访问软构件提供的一致服务。

参考文献:

- [1] 全春来等. 并行工程集成框架 CEIF 总体设计 [J]. 软件学报, 1998, 9 (suppl), 197-201.
- [2] Idalina V, et al. Efficient communication in a design environment [A]. In: P. Pennino ed, 33rd Design Automation Conference, Las Vegas, IEEE Computer Society Press, 1996, 169-17.
- [3] Schermann B. Modeling Design Tasks and Tools-The link between Product and Flow Model [A]. In: 34th Design Automation Conference, Anaheim, IEEE Computer Society Press, 1997, 564-139.
- [4] 郭阳, 杨军, 陈旭灿, 李思昆. 基于设计体模型的产品数据管理技术 [J]. 国防科技大学学报, 1998, 20 (3), 53-56.

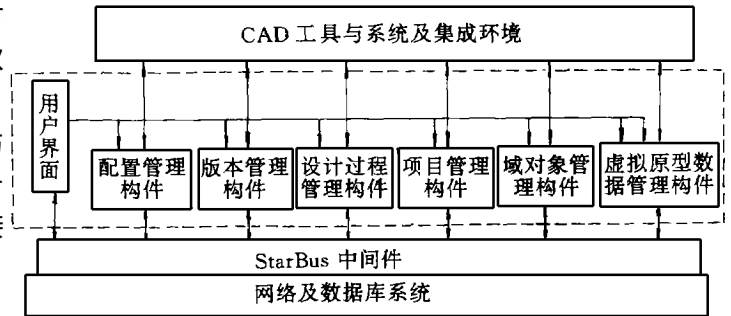


图 1 YG-bridge 集成框架的结构

Fig. 1 Architecture of YG-bridge framework