

面向复杂产品数据管理的适应性模型研究^①武园浩^② 田 凌

(清华大学精密仪器与机械学系 北京 100084)

摘 要 为了能够满足复杂产品研发过程中的不断迭代变化的产品数据管理需求,建立了面向复杂产品数据管理的适应性模型(APDM),给出了该模型的元模型的形式化定义,在此基础上建立了该模型的以产品结构为中心的领域模型表达框架,并以产品结构管理和文档版本演化为例分析证明了该模型的良好表达能力与灵活性。设计了复杂产品适应性数据管理系统的体系结构以及基于可扩展标记语言(XML)格式的中性文件的异构系统产品数据交换接口,完成了适应性产品数据管理系统的开发,并通过在集成电路装备研发中的应用,验证了该模型的正确性与有效性。

关键词 复杂产品, 数据管理, 适应性模型, 可扩展标记语言(XML), 异构系统接口

0 引 言

复杂产品的复杂性不仅体现在产品的组成复杂、技术复杂,同时体现在其客户需求复杂、项目管理过程复杂等方面^[1]。产品数据管理技术能够为复杂产品研发过程中的数据管理、文档管理等需求提供有效支持^[2]。文献[3-5]从产品数据管理系统的功能、体系结构、数据模型等方面做了研究,这些研究主要通过面向对象技术进行产品数据建模,由于模型中数据对象的属性、操作与数据对象之间的关联关系处于代码级别的紧耦合状态,当业务逻辑发生变化时,系统修改成本很高。一些学者针对此问题进行了研究。文献[6]提出了业务基元的概念,以满足数据管理的灵活性。业务基元是指包含一类设计对象以及与之相关的属性和业务操作的单元,其包含的内容可以进行自由配置。文献[7]从广度、深度、粒度三个维度全面描述了产品的生命周期信息,在此基础上从版本和映射两个角度研究了产品的动态演化过程。文献[8]研究了通过领域纲要与领域模板定义元模型,以及动态实例化生成应用模型的方法,以实现模型驱动下的系统快速开发。文献[9]结合代理技术与模型驱动软件架构,研究了如何提高系统的适应性与扩展能力。文献[10]

结合元模型建模技术,研究了可定制的产品数据模型。复杂产品研发过程中不同设计人员可能使用不同的系统,这对产品数据管理又提出了异构数据共享与集成的需求。在这方面,文献[11]将元数据建模与本体结合,通过产品数据本体保证产品数据的一致性;文献[12]研究构建了网络化协同设计平台,使用标准数据格式进行异构系统集成;文献[13]则研究了基于单一数据源的飞机产品工程数据集成。本研究在以上研究的基础上,建立了面向复杂产品数据管理的适应性模型(adaptive model for complex product data management, APDM),该模型具有良好的表达能力与灵活性,能够快速响应复杂产品数据管理需求变化,提高产品数据管理系统的适应性。本文对 APDM 以及基于该模型设计的适应性产品数据管理系统的系统结构、基于可扩展标记语言(extendable markup language, XML)格式的异构系统数据交换接口进行了阐述与分析,并给出了所开发的系统在集成电路装备研发团队的应用实例。

1 APDM

传统产品数据管理系统通常建立在面向对象的模型基础之上,数据实体由类作封装,类中的属性反

① 国家自然科学基金(51175287)和国家重大科技专项(2011ZX02403)资助项目。

② 男,1987年生,博士生;研究方向:产品数据管理,知识工程,系统集成技术;联系人,E-mail: wyh35071208@126.com
(收稿日期:2012-04-10)

映了实体的特征,类中的操作反映了业务规则。基于这种建模手段,大量领域知识被封装到系统的代码中,修改成本很高。适应性对象模型^[14,15]对类、属性和关系等通过元模型进行进一步抽象,并在元模型的基础上建立领域模型,元模型和领域模型记录在数据库或文件系统中,在运行时通过解释机制将模型动态实例化。这样当业务需求发生变化时,通过配置领域模型即可以实现系统的动态修改,而不需要重新部署。本节借鉴了适应性对象模型的设计思想,首先给出了 ADPM 的元模型,然后在元模型的基础上建立了 ADPM 的领域模型,并对 APDM 模型的性能进行了分析。

1.1 ADPM 元模型

元模型是对模型域的抽象,也是建立领域模型的基础。复杂产品数据管理的适应性模型的元模型包括 4 个基本类:实体类、属性类、关联类、操作类。

(1) 实体类。实体类是对具体产品数据的抽象,产品零部件、CAD 文档、仿真分析文档、工艺文档等都是实体类的实例。实体类定义如下:

$$Entity = \langle EnID, Set\{ (Att, V_{att}) \}, Set\{ Rel \}, Set\{ Oper \} \rangle \quad (1)$$

其中 $EnID$ 为实体对象的唯一确定标识; $Set\{ (Att, V_{att}) \}$ 为二元组 (Att, V_{att}) 的集合,其中 Att 表示属性对象, V_{att} 为属性对象的值; $Set\{ Rel \}$ 为实体中包含的关系对象的集合; $Set\{ Oper \}$ 表示实体中包含的操作对象的集合。属性、关系以及操作类的定义将在下文阐述。

(2) 属性类。属性是对实体特性的描述。产品零部件的材料、性能参数, CAD 图文档的创建时间、版本、图号等,都是属性类的实例。属性类的定义如下:

$$Att = \langle AttID, AttName, AttValueType, Domain \rangle \quad (2)$$

其中 $AttID$ 为属性对象的唯一确定标识; $AttName$ 为属性对象的名称; $AttValueType$ 为属性对象的取值类型,如整型、字符串、浮点型等; $Domain$ 为属性对象取值的有效范围。

(3) 关联类。关联表示实体之间的联系,零部件之间的装配关系、零部件实体与图文档的依赖关系等都是关联类的实例。在传统面向对象模型中,实体之间的关联关系通过一个实体对象作为另一个实体的数据成员体现,是一种紧耦合的实现方式,本文中的元模型将关联作为独立的对象加以定义,完成实体的松散耦合与灵活配置。关联类的定义如下:

$$Rel = \langle RelID, RelType, subject, Set\{ object \} \rangle \quad (3)$$

其中 $RelID$ 为关联对象的唯一确定标识; $RelType$ 为关联类型的语义描述,如版本关联类型、聚合关联类型等; $subject$ 和 $object$ 分别为关联关系的主体和客体,它们均是实体类对象实例,关联关系的客体可以为多个,故用集合的形式描述。 $Set\{ object \}$ 为非空集合,其包含的元素个数大于 1 时,表明为一对多的关联类实例,如零部件的聚合关联,因为一个部件可以由多个部件或零件组成;当其包含元素个数为 1 时,标明为一对一的关联类实例,如零部件的网格划分文档与零部件的仿真结果之间的关联必须是一一对应的。

(4) 操作类。操作类是对产品数据实体的操作的抽象,更新文件版本、配置产品结构都是操作类实例。操作类的定义如下:

$$Oper = \langle OperID, Input\{ object \}, Output\{ object \}, OperImpl \rangle \quad (4)$$

其中 $OperID$ 为操作类对象的唯一确定标识; $Input\{ object \}$ 与 $Output\{ object \}$ 分别操作的输入输出对象集合,这里的对象可以是上述实体类、属性类以及关联类的实例; $OperImpl$ 为操作的实现。

上述的元模型中实现了实体与属性、实体与关联、实体与操作的分离。将属性、关联与操作以集合的形式装入实体中,具体实体对象中包含的属性、关系与操作都可以在配置文件中进行动态配置并在运行时解释执行,从而避免了修改源代码,能够更快适应业务的变化,满足复杂产品数据管理的需求。

在复杂产品开发过程中,往往需要多个设计主体的协同合作,每个设计主体的关注点是不同的,为此引入“切面”的概念来描述产品不同类型的信息。切面是设计主体感兴趣的某一类实体和与其相关的关联类对象与操作类对象的集合,产品的需求定义、仿真信息、工艺信息等都是产品的不同切面。切面的定义如下:

$$Aspect = \langle Set\{ En \}, Rel(Set\{ En \}), Oper(Set\{ En \}) \rangle \quad (5)$$

其中 $Set\{ En \}$ 表示设计主体关注的实体对象集合; $Rel(Set\{ En \})$ 和 $Oper(Set\{ En \})$ 分别是与集合中实体相关的关系类对象与操作类对象。

1.2 APDM 领域模型的建立与分析

在元模型定义的基础上,建立面向复杂产品数据管理的领域模型。从信息变化的角度来看,产品的设计过程是以产品结构为中心的设计信息不断丰

富的过程,故以产品结构作为 APDM 领域模型的组织所有工程数据与文档的中心。产品零部件实体之间通过聚合关联类实例组成产品结构树,产品的功能定义文档、CAD 模型文件、仿真分析结果、工艺加工文档、供应商信息等文档实体都通过依赖关联类实例与产品结构树的零部件实例产生关联,不同类型的文档实体与数据组成产品数据的不同切面,同时切面中的数据实体通过关联类进行映射与交互。APDM 领域模型的整体框架如图 1 所示。

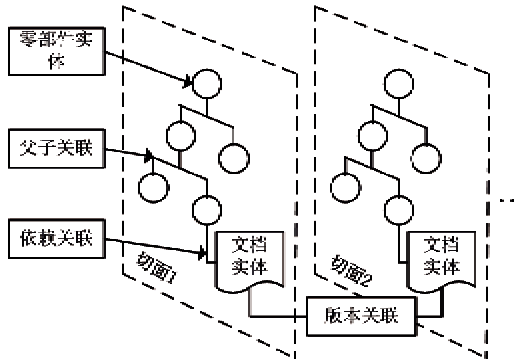


图1 APDM 领域模型框架

在 APDM 领域模型中,最重要的两类实体模型是产品零部件实体和文档实体。下面通过产品结构管理和文档的版本管理两个角度分析 APDM 模型的性能。

产品结构管理是产品数据管理中一项重要内容,在复杂产品的研发过程中,产品结构树处于不断演化之中。在 APDM 中,产品结构树通过零部件实体对象与聚合关联类对象组成,产品结构管理在 APDM 中表现为对聚合关联类对象中关联客体集合中包含的零部件实体的操作。定义聚合关联类为 $Rel = \langle RelID, RelType = 'COMPOSE', subject, Set(object) \rangle$, 其中 $subject$ 表示装配关系中部件实体, $Set(object)$ 为装配关系中组件实体的集合。如图 2 所示, 部件 A 由零件 A 和零件 B 构成, 其聚合关联类中客体集合包含两个元素, 即 $Part_A$ 和 $Part_B$, 当为部件 A 添加零件 C 时, 只需要修改原聚合关联类中的客体集合, 将零件 $Part_C$ 放入 $Set(object)$ 中即可。由于将零部件实体与其之间的聚合关联解耦, 调用操作类修改上述聚合关联类时, 对零部件实体本身或其他相关数据不需要做任何改变。

文档版本反映了产品数据的动态演化过程, 文档的版本演化规则一般有两种: 线型模型与树状模

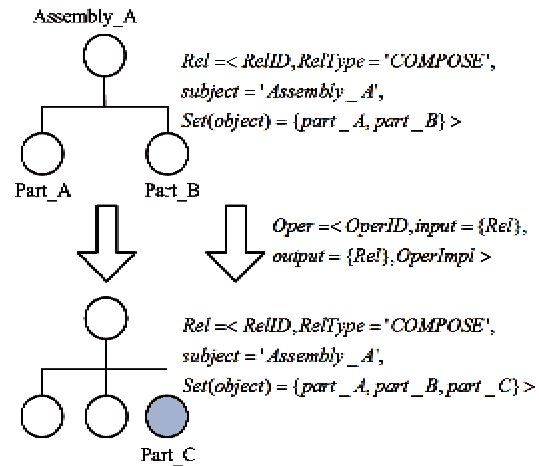
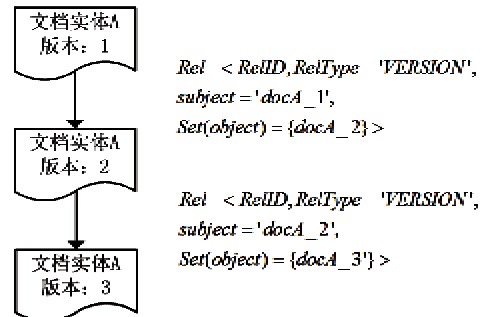
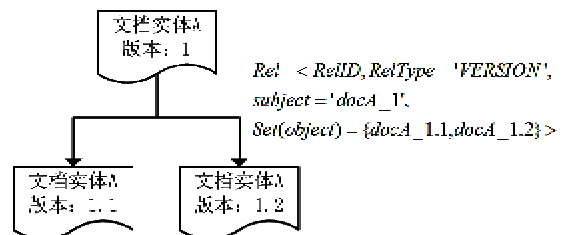


图2 产品结构管理的 APDM 描述

型。线型模型是指一个文档的版本有至多一个前驱版本和至多一个后继版本, 没有前驱版本的文档为初始文档, 没有后继版本的文档为最终文档, 其他文档均为中间版本文档。如果规定同一个文档可能派生出多个后继版本, 这时版本演化规则将变为树状模型。在 APDM 中, 使用版本关联类来完成文档版本的管理与控制。定义版本关联类为 $Rel = \langle RelID, RelType = 'VERSION', subject, Set(object) \rangle$, 其中 $subject$ 实体表示低级版本, 而 $Set(object)$ 表示该 $subject$ 的后继高级版本集合, 当 $Set(object)$ 被约束为单元素集合时, 版本管理规则为线性模型, 而当 $Set(object)$ 可以包含多个实体元素时, 版本演化将演化为一个树状模型, 如图 3 所示。



(a) 版本演化的线型模型



(b) 版本演化的树状模型

图3 版本规则的 APDM 描述

2 基于 APDM 的系统结构

基于 APDM,本文设计了面向复杂产品的数据管理系统的体系结构,如图 4 所示。

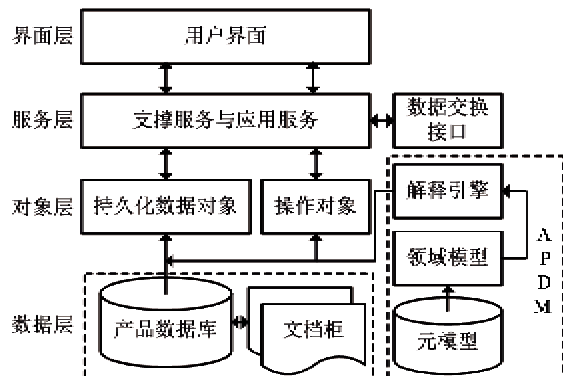


图4 基于 APDM 的复杂产品数据管理系统体系结构

系统结构分为数据层、对象层、服务层、界面层以及 APDM 模块与异构系统数据交换接口模块两个模块。数据层中包括产品数据库与文档柜,存放产品的所有工程数据与文档。APDM 模块包含元模型,领域模型和解释引擎三部分。其中领域模型是基于元模型定义并针对具体的业务而构建的数据模型,是对元模型的实例化。复杂产品数据管理的领域模型包括零部件实体、文档实体、聚合关联、依赖关联、版本关联、各种属性类以及操作类等定义。解释引擎则根据定义好的领域模型在运行时完成模型的实例化,并结合产品数据库中的数据生成数据对象与业务操作对象,它们构成了系统的对象层。服务层包括支撑服务与应用服务,界面层负责调用相应服务并与用户交互。

基于 APDM 的系统架构与一般的分层系统架构的主要区别在于,其数据对象与业务操作对象构成的对象层并不是对产品数据的直接抽象,而是基于产品数据由 APDM 的解释引擎将领域模型动态实例化生成。当需求发生变化时,领域专家可以快速修改与配置领域模型,并通过解释引擎重建持久化对象,快速满足复杂产品数据管理的变化需求。

复杂产品研发通常为多个设计主体的协同合作,各设计主体之间可能使用不同的设计系统。为了满足异构系统中产品结构数据能够共享,使产品结构作为组织所有的工程数据和文档的中心数据,本文设计了基于 XML 中性文件的异构系统的数据接口。如图 5 所示,用户系统通过 XML 转换器将自身产生的数据转换成 XML 格式的中性文件,系统通

过 XML 转换器将 XML 中性文件转换成结构化数据并存入数据库中,XML 转换器必须满足事先定义好的 XMLScheme 文件规范,以保证 XML 中性文件中数据的正确性与完备性。

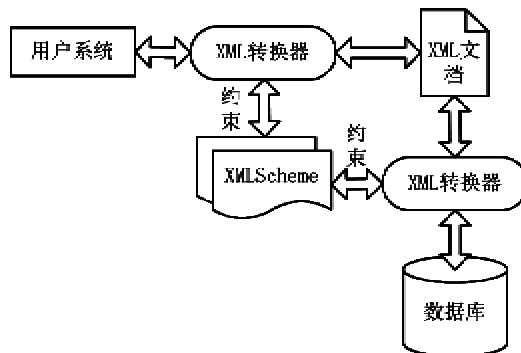


图5 异构系统产品数据交换接口

3 工程应用

集成电路(integrated circuit, IC)装备是典型的工业复杂产品^[16],已经成为一个国家科技发展的制高点之一。IC 装备的研发受到国家重大科技专项的支持,正处于快速发展阶段,但由于技术并未成熟,数据管理需求在研发的不同阶段呈现不同的状态,另一方面,IC 装备研发过程需要多领域多学科的研究人员参与,其数据管理必须满足不同领域异构系统的集成需求。本文开发了面向 IC 装备研发的适应性产品数据管理系统,并应用于某 IC 装备研发团队。系统以 J2EE 作为技术支持平台,结合 Struts2、Spring、Hibernate 等开源框架、XML 序列化与反序列化工具、工作流引擎等技术开发。

首先根据 IC 装备研发团队的具体数据管理需求定制其初步领域模型。构建数据实体类对象,包括产品对象、部件对象、零件对象、设计文档对象、仿真文档对象、订单对象、外协外购对象,等等;分析需要管理的各数据实体对象的属性,构建材料、日期、名称、地址等等属性对象;构建关联类对象,如构建聚合关联类来完成产品结构树建立,构建依赖关联类以描述设计文档与零部件的依赖关系,构建版本关联类以表示文档的版本演进关系;构建对以上数据对象、属性对象、关联对象的操作类对象,如版本操作,产品结构配置操作等;最后构建不同的切面对象,为不同用户提供不同的视图,如图文档管理视图、仿真信息视图等。本文构建的领域模型使用 XML 文件定义,解释引擎基于 XML 解析技术与 java

的反射机制开发,在系统运行时完成 XML 领域模型的动态实例化。领域模型中零部件实体对象以及其引用的材料属性对象与聚合关联对象的 XML 定义片段如图 6 所示。



图 6 零部件实体的领域模型定义

由于产品的复杂性,IC 装备研发中的数据管理需求处于不断迭代之中。相比于传统的数据管理系统,本文开发的适应性产品数据管理系统能够更好地响应需求的变化,降低系统实施成本。系统的适应性包含三个层次:(1)元模型的适应性。系统使用的元模型中将数据实体、属性、操作、关联关系解耦,当对数据实体进行操作时,不会对其他对象产生影响,如产品结构配置,只需要调用相应的操作修改聚合关联对象,而产品数据本身不需要发生变化。(2)可配置领域模型的适应性。系统的领域模型通过外部的配置文件保存,系统功能需要修改时,只需要对配置文件进行相应修改,通过解释引擎重建数据对象,系统无需再次部署即可生效。例如添加新的文档类型,扩充材料属性的可选项等,这些操作制度要修改领域模型的 XML 配置文件。(3)可配置切面提供的适应性。切面类定义了一类实体及其相关的关联、操作等。不同的切面可以为关注点不同的设计人员提供不同的数据管理视图。切面类在配置文件中保存,当设计人员的视图需求发生变化时,如需要增加查看外协订单的视图,则可以动态新建外协订单切面,令其包含的外协订单文档实体及其相关对象,可快速为设计者提供所需视图。

IC 装备研发团队的结构设计工程师主要使用 Solidworks 软件进行建模。为了和其他设计仿真及工艺软件完成数据交互与集成,根据图 5 所示的异构系统产品数据交换接口设计方案,实现了 Solid-

works Workgroup PDM 与本系统的产品结构数据集。在此基础上,本系统以产品结构树为中心,完成了产品数据管理、文档管理、仿真信息管理等数据管理需求。图 7 中右上方为设计人员在 Solidworks Workgroup PDM 中模型的装配结构树,通过基于 XML 异构系统接口导入至本系统,并通过产品结构树管理各个零部件的需求定义文档。

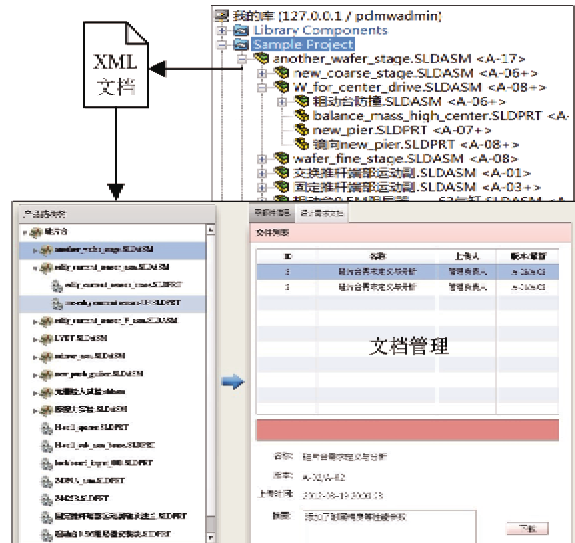


图 7 Solidworks Workgroup PDM 的系统接口及文档管理实现

4 结 论

本文结合适应性对象建模思想,建立了面向复杂产品数据管理的适应性模型(APDM),给出了其元模型定义与领域模型框架,并开发了基于该模型的产品数据管理系统。对模型的分析表明其相比于面向对象具有更强的灵活性与表达能力,并通过在 IC 装备这一典型工业复杂产品研发中的应用,证明了通过将领域模型定义独立于程序代码,能够快速重建数据对象,及时满足 IC 装备研发中不稳定的产品数据管理需求。同时基于 XML 的产品数据交换接口能够完整实现异构系统的产品结构数据交互,系统可以构建单一的产品数据树,并通过将不同类型的数据实体与产品结构中的零部件关联,完成了 IC 装备研发中的产品数据管理任务。

参考文献

- [1] 范文慧,肖田元. 复杂产品协同设计、仿真、优化一体化平台. 科技导报,2007,25(4):15-24
- [2] 李建明,童秉枢,许隆文. 产品数据管理技术的现状与

- 发展. 计算机集成制造系统, 1998, (6): 1-6
- [3] 刘长明. PDM 关键技术的研究与开发: [硕士学位论文]. 长春: 吉林大学计算机科学与技术学院, 2011. 33-38
- [4] Sudarsan R, Fenves S J, Sriram R D, et al. A product information modeling framework for product lifecycle management. *Computer-Aided Design*, 2005, 37(13): 1399-1411
- [5] 曹健, 黄越, 张申生. 支持协同产品开发的产品数据管理模型研究. 高技术通讯, 2002, 12(3): 79-82
- [6] Feng G, Cui D L, Wang C G, et al. Integrated data management in complex product collaborative design. *Computers in Industry*, 2009, 60(1): 48-63
- [7] 武建伟, 江伟光, 潘双夏. 面向产品全生命周期的集成模型及演化研究. 计算机集成制造系统, 2009, 15(8): 1493-1499
- [8] 耿俊浩, 张振明, 田锡天等. 面向产品生命周期管理的特定领域建模方法. 计算机集成制造系统, 2008, 14(2): 262-267
- [9] Liang X, Des G. Adaptive agent model: software adaptivity using an agent-oriented model-driven architecture. *Information and Software Technology*, 2009, 51(1): 109-137
- [10] 刘刚, 高琦, 康延钦等. 基于元模型的可定制产品数据管理模型研究. 组合机床与自动化加工技术, 2010, (7): 97-101
- [11] 顾巧祥, 祁国宁, 纪杨建等. 基于元数据的产品数据本体建模技术. 浙江大学学报, 2007, 41(5): 736-741
- [12] Tian L, Chen J Z, Wang Q Y, et al. CoDesign Space: a collaborative design support system in a network environment. *International Journal of Computer Integrated Manufacturing*, 2007, 20(2): 265-279
- [13] 卢鹤, 于勇, 杨五兵等. 飞机单一产品数据源集成模型研究. 航空学报, 2010, 31(4): 836-841
- [14] Yoder J W, Balaguer F, Johnson R. Architecture and design of adaptive object-models. *ACM SIGPLAN Notices*, 2001, 36(12): 50-59
- [15] Yoder J W, Johnson R. The adaptive object-model architectural style. In: *Proceedings of the 3rd Working IEEE/IFIP Conference on Software Architecture*, Montreal, Canada, 2002. 3-27
- [16] Su X S, Tian L. A flexible card configuration system for testing of instruments in wafer fabrication equipment. In: *Proceedings of 2011 International Conference on Electric Information and Control Engineering*, Wuhan, China, 2011. 5625-5630

Research on adaptive model for complex product data management

Wu Yuanhao, Tian Ling

(Department of Precision Instruments and Mechanology, Tsinghua University, Beijing 100084)

Abstract

In order to satisfy the changing data management demand during the complex product development, an adaptive model for complex product data management, called APDM for short, was proposed. The formal definition of the APDM's meta-model was presented, and based on this, a product structure centered representation framework for its domain model was built. The good flexibility and representation ability of the adaptive model were verified by the analysis done through the examples of product structure management and document version evolution. A layered system architecture based on the adaptive model together with a product data exchange interface based on the extendable markup language (XML) neutral document were also design. An adaptive product data management system was developed, and its application in integrated circuit equipment research proved the effectiveness of the adaptive model and the system.

Key words: complex product, data management, adaptive model, extendable markup language (XML), heterogeneous system interface