

基于 XML 的信息集成系统的研究与实现*

刘科研¹, 万丽荣², 曾庆良², 范文慧³

(1. 北京航空航天大学 自动化学院, 北京 100083; 2. 山东科技大学, 山东 泰安 271019; 3. 清华大学 自动化系, 北京 100084)

摘 要: 研究了基于 XML 的共享信息的表达, 通过共享信息模型实现了异构分系统间的信息传递和共享。最后结合实际 CAPE 项目, 以统一数据访问平台为例, 对以上技术的可行性进行了验证。

关键词: XML; 信息集成; 信息模型

中图法分类号: TP391 文献标识码: A 文章编号: 1001-3695(2005)04-0149-03

Research and Realization of Information Integrated System Based on XML

LIU Ke-yan¹, WAN Li-Rong², ZENG Qing-liang², FAN Wen-hui³

(1. Dept. of Automation, Beijing University of Aeronautics & Astronautics, Beijing 100083, China; 2. Shandong University of Science & Technology, Taian Shandong 271019, China; 3. Dept. of Automation, Tsinghua University, Beijing 100084, China)

Abstract: Study the information representation based on XML and achieve the information transfer and share between different isomerous systems using the sharing information model. An application example about CAPE(Concurrent Art-to-Product Environment) project is provided, and the validity of the methods above is proved.

Key words: XML; Information Integration; Information Model

信息技术的发展为人们提供了数不胜数的软件系统, 这些软件分别面向不同领域、不同专业, 解决不同问题。没有一个大一统的软件能够覆盖国民经济不同行业、不同层次的需求, 这一现实要求人们能提供将异质、异构系统整合为一个集成系统的技术。系统集成已成为数字化工程的核心技术, 也是当前研究热点当中的热点^[1]。

事实上, 实现系统集成的关键技术只有两个: 软件互操作和信息集成。CORBA 和 DCOM 为软件互操作提供了标准, 并已得到广泛支持。信息集成的困难是构造集成平台的重大障碍。要解决这一问题, 就需要对其进行深入研究。

产品信息集成在技术上可分为四个层次: 消息通信、产品数据共享、互操作及数据与知识重用。笔者比较了信息集成的方法, 采用 XML 进行系统间的数据传递, 由信息集成系统进行数据的统一处理, 确保多个系统可以共享数据, 从而达到共享要求。某部为提高企业的信息化水平, 降低产品开发成本、缩短开发周期, 针对某一型号产品研发过程, 实施 CAPE (Concurrent Art-to-Product Environment) 预研项目计划。我们在实现其内部信息集成工作的设计和研究的过程中, 采用 XML 技术作为信息集成过程中的信息建模和信息交换、共享过程中的支撑技术。

1 当前的信息集成方法

1.1 基于数据库的信息集成

两个数据库之间的数据交换通过直接读取对方数据库或

通过数据库映射来获取对方数据。这种方法的优点是实现比较简单, 像 ODBC, JDBC 都是比较成熟的技术, 实现的成本也比较低。缺点是系统必须在数据库中存储所要交换的数据, 并提供相应的数据存取方法; 一般只能交换静态的数据, 对于程序运行中存在的动态数据是无法进行交换的。这种方法集成时, 两个系统之间都得开发不同的接口, n 个系统之间的集成必须开发 $n \times (n - 1)$ 个接口, 接口的重用性比较差。

1.2 基于 PDM 的信息集成

基于 PDM 的信息集成的中心思想是围绕产品开发过程组织数据, 使产品数据管理与数据使用紧密相关, 是理想的企业信息集成方法。但是当前的 PDM 系统提供的模型管理粒度较大, 只能基于文档级, 不能深入文档内部, 无法满足产品信息在不同粒度上的共享。

1.3 基于 STEP 的信息集成

产品数据表达与交换标准 STEP(Standard for the Exchange of Product Model Data) 是 ISO 提出的产品数据表达与交换的国际标准。STEP 是一套关于产品整个生命周期内产品数据的表达与交换的国际标准。它提供了一种不依赖于具体系统的中性机制, 支持产品信息在不同的计算机应用系统之间集成与交换的完整性和一致性。它不仅适合于中性文件交换, 而且是实现共享产品数据库及产品数据的长期存档的基础。

基于 STEP 集成的缺点就是 STEP 要求应用程序必须遵循共同的应用协议(AP), 而应用协议还远远不够完善并且应用协议之间没有实现标准化, 再加上目前可供借用的 STEP 中性文件开发工具比较少, 因此采用 STEP 标准描述共享信息模型, 实现产品信息集成仍有一定的难度。

2 基于 XML 的信息集成

XML 技术起源于 Web 中的半结构化的标记语言, W3C 联合组织制定了一系列 XML 标准规范, XML 得到了业界的支持和响应, XML 极具成为新一代数据交换标准的潜力, 其精髓在于允许文档的编写者制定基于信息描述、体现数据之间逻辑关系的自定义标记。XML 采用半结构化数据模式, 数据模式可动态改变, 可以随时加入新的实体, 也能满足数据语义复杂、数据粒度小的要求, 是理想的信息建模语言。XML 可以支持多种数据源, 如纯文本文档和关系数据库, XML 纯文本文档是最基本的、也是最简单的, 将数据存储于文件中可以方便地读取。XML 文件可以同关系数据库建立映射, 转换为表结构存储于数据库中。另外目前已经有不少功能强大而且成熟性强的 XML 支持工具。

XML 作为一种可扩展性标记语言, 其自描述性使其非常适用于不同应用间的数据交换和异构数据的集成, 同时具备很强的开放性和可扩展性, 具有广阔的应用前景。另外, XML 还允许为特定的应用领域制定特殊的数据格式, 在领域内各系统间通过传递格式化数据实现数据的交换与共享。所以 XML 不仅为 Web 数据带来了结构化、智能化和互操作性, 而且可以作为应用之间转换和传送数据的有效手段。

采用 XML 进行文件交换, 在实现上远比采用 STEP 中性文件进行传送简单, 因为 XML 开发程序工具提供了大量的处理 XML 文件的功能。相比较而言, 采用 STEP 中性文件进行传送, 可供借用的开发工具只有美国 STEPTools 公司提供的 ROSE 工具(不是 Rational Rose) 或另一个德国产品 ProStep。这两个产品无论是在成熟程度还是功能上均不如 XML 支持工具^[2]。虽然 XML 文档具有数据存储管理的基本功能, 但是数据库管理系统比 XML 具有更加强大的数据管理功能: 数据存取的高效性; 数据的一致性自动化保证机制; 强大的数据完整性保证机制; 多用户并发访问控制机制。所以目前在实际应用中, 大量数据的存储管理还是依靠数据库管理系统, XML 的核心作用主要体现于共享数据的交换实现。

2.1 系统间信息集成

2.1.1 XML 信息模型

XML 文件包括文档类型定义 DTD 和 XML 文档。文档类型定义 (DTD) 是一套关于标记符的语法规则, 它告诉你可以在文档中使用哪些标记符、它们应该按什么次序出现、哪些标记符可以出现于其他标记符中、哪些标记符有属性等。它可以是 XML 文档的一部分, 但是它通常是一份单独的文档或者一系列文档^[3]。文档类型定义 DTD (Document Type Definition) 用于规定文档的逻辑结构, 定义页面的元素、元素的属性以及元素的属性之间的关系。

CAPE 环境的共享信息模型采用 XML 语言描述, 由一系列 DTD(文档类型定义) 和 XML 文档组成。模型按照分系统来组织, 分为设计分系统信息模型(设计信息模型)、仿真分系统信息模型(仿真信息模型)和管理分系统信息模型(管理信息模型) 模型等。

2.1.2 共享模型信息的表达

系统间的共享信息模型是基于 XML 的, 由一系列文档类

型定义 (DTD) 和对应的 XML 文档组成。XML 文档是对应 DTD 的数据值定义。XML 文档首先是格式良好的, 它必须遵循 XML 1.0 规范中的语法规则; 另外, 它还要是一个有效的 XML 文档, 必须遵循 DTD 中声明的种种规定。每个 XML 文档对应一个 DTD, 而一个 DTD 可以对应多个 XML 文档。DTD 是数据的语义定义, XML 文档则是数据值的定义。DTD 对于 XML 文档如同数据表中的表结构对于表中的数据值。

以下是设计分系统中“设计指标”这一数据对象的数据定义(表 1) 和由 XML 编辑器 XML SPY 软件生成的设计指标 DTD 和对应的一个 XML 文档。Device_type_n 表(表 2) 的命名中, n 与表 1 中的编号相关联, 表示设计方案 n 中所采用的设备类型集合。设计指标 DTD 描述如图 1 所示。

表 1 设计指标表(design_specification)

编号	描述	反应时间	解算精度	发现距离	可靠性	基阵噪声	发射深度	处理匹数
1		25 min	10	10km	5h	118 db	200 m	10
2		20 min	15	8km	10h	100 db	250 m	8

表 2 选用设备类型表(device_type_n)

名称	鱼 A	沉 A	X 声纳
类型	YL(鱼雷)	SL(水雷)	TCSB(探测设备)



图 1 设计指标 DTD 标定

对应的一个 XML 文档表示如下:

```
<?xml version = "1.0" encoding = "GB2312" ?>
<!DOCTYPE 设计方案 SYSTEM "D:\DTD\2.dtd">
<设计方案>
  <table_design_specification>
    <编号 唯一 = "是">xml001</编号>
    <描述>test-xml001</描述>
    <反应时间>15</反应时间>
    <解算精度>10</解算精度>
    <发现距离>10 km</发现距离>
    <可靠性>10h</可靠性>
    <基阵噪声>118db</基阵噪声>
    <发射深度>200m</发射深度>
    <处理匹数>10</处理匹数>
  </table_design_specification>
  <table_device_type 编号 = %xml001 "唯一 = 是">
    <名称>鱼 A</名称>
    <类型>鱼雷</类型>
  </table_device_type>
  <table_device_type 编号 = %xml001 "唯一 = 是">
    <名称>沉 5</名称>
    <类型>水雷</类型>
  </table_device_type>
  <table_device_type 编号 = %xml001 "唯一 = 是">
    <名称>导 4</名称>
    <类型>导弹</类型>
  </table_device_type>
</设计方案>
```

2.2 模型信息的共享

CAPE 环境的信息模型采用 XML 语言描述, 以文件形式

存放在文件服务器上; 数据服务器负责文档信息的管理; 应用系统通过客户端调用平台提供的服务来提交/读取 XML 文档, 实现系统间数据的共享。

系统间模型信息的共享通过发布和订购文档类型定义 (DTD) 和 XML 来实现。需要实现数据共享的应用系统首先向平台注册自身的信息, 主要包括两部分内容:

- (1) 发布 (Publish)。向平台提供该系统可以提供的文档类型定义 (DTD) 及其 XML。
- (2) 订购 (Subscribe)。向平台申请该系统需要读取的其他系统发布的文档类型定义 (DTD) 及其 XML。

系统注册信息过程如下: 需要信息共享的系统向平台提交该系统提供的 DTD, 然后向平台提交对应 DTD 的 XML 文档。其他系统可以通过该平台申请需要共享的该分系统提供的 DTD, 并且能够通过平台客户端获取该 DTD 和 XML 文档。这两部分信息模型分别在平台的 Publish 和 Subscribe 下显示。

参加集成的各个分系统使用发布和订购完成了系统间数据的交互和信息的传递, 主要过程如下:

- 以 A, B, C 三个系统为例,
- (1) A 接收到 B 的数据模式要求 DTD;
 - (2) A 根据 B 的数据模式要求 DTD 生成 XML 文档, 并且发送给 B;
 - (3) B 读取 XML 文档中的数据, 进行应用处理;
 - (4) B 将应用处理的结果写入本地数据库中;
 - (5) B 将应用处理的结果按照 C 的数据模式要求 DTD 生成 XML 文档, 并且发送给 C。

3 统一数据访问平台的实现

某部为提高企业的信息化程度、降低产品开发成本、缩短开发周期, 针对某一型号的产品实施 CAPE 预研项目计划。CAPE 环境由多个应用分系统组成, 包括 CAD 分系统、专家评估分系统、文档辅助生成系统、仿真分系统等。这些分系统都是针对特定应用领域而设计的, 各个分系统采用的数据结构不同, 分系统间的数据交互困难, 目前仍是一个个的“信息孤岛”^[4]。但另一方面, 这些系统间还存在着大量的产品数据交换和共享的需求, 如专家评估系统需要对 CAD 的设计结果进行评估; 文档辅助生成系统需要得到其他各个分系统的结果数据用于生成标准化文档; 仿真分系统需要得到 CAD 分系统的设计结果进行仿真分析等。如图 2 所示, 其中虚线部分是一个重复的循环过程。为了使分系统间协同工作, 我们采用 XML 语言进行信息建模, 各个分系统通过统一数据访问平台 (Unified Data Access Platform) 实现系统间产品信息的集成。

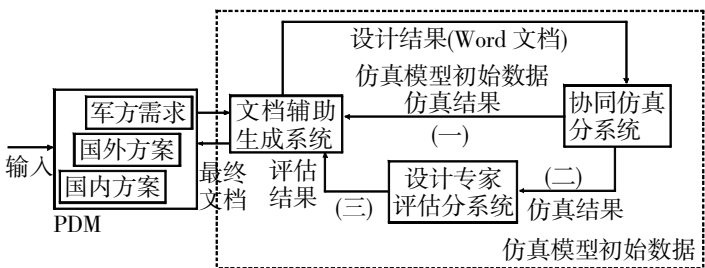


图 2 CAPE 环境信息流程图

统一数据访问平台是对 CAPE 环境共享信息模型进行管

理、维护, 并为 CAPE 环境实现信息集成提供支持的数据平台。平台采用 C/S 结构, 即由客户端和文件服务器、数据库服务器两层结构组成。CAPE 环境的信息模型采用 XML 语言描述, 以文件形式存放在文件服务器上; 数据库服务器负责文件信息的管理; 应用系统通过客户端调用平台提供的服务来提交/读取 XML 文档, 实现系统间数据的互访。我们采用 C++ Builder 6.0 作为开发语言, 数据库采用 Oracle 9, 文件服务器采用 Server-U, 用 DOM 接口实现对 XML 文档的解析。图 3 是统一数据访问平台的应用界面。



图 3 统一数据访问平台主界面

该平台提供的服务包括以下几个功能模块: 主界面模块、服务器连接模块、管理员登录模块、分系统模型管理模块、DTD 的管理模块、XML 文档管理模块、关系管理模块、查找模块和系统设置模块。平台的八个功能模块共同提供的服务可概括为如下几个方面:

- (1) 发布系统信息。任何一个新的分系统添加到平台中, 首先要将其提供的数据和系统本身的信息添加到平台中, 具体由添加分系统、添加 DTD、添加 XML 文档等操作来实现。
- (2) 订购系统所需共享信息。有共享需求的分系统还必须向平台添加其共享数据的信息来申请其他分系统提供的数据, 具体由添加系统间关系、添加关系 DTD 等操作来实现。
- (3) 管理、维护 CAPE 环境的信息模型。当某个分系统的共享信息或者需求信息发生改变时, 甚至某个分系统不再使用时, 可以调用平台服务相应地修改该分系统提供的信息模型和共享需求信息, 具体由一些修改、删除操作来实现。
- (4) 查找并获取需求数据。分系统可以通过平台查询需求的 DTD 和 XML 文档, 并且通过平台获取该 DTD 和文档。具体由查找操作来完成, 然后通过文档另存或者直接从服务器上下载来获取文档。
- (5) 操作关系数据。其他分系统提供的、该系统有共享需求的这一类数据信息在平台中称为分系统的关系数据或称为分系统订购的数据。在平台中提供关系数据的分系统称为该分系统的关系系统, 描述关系数据的 DTD 称为该分系统的关系 DTD。平台对关系数据的操作由关系管理模块来完成, 具体操作有添加分系统的关系系统、关系 DTD 以及删除关系系统和关系 DTD 等。在添加关系数据时首先添加关系系统, 然后再添加关系系统提供的 DTD。

统一数据访问平台主要完成对 CAPE 环境中共享信息模型的管理、维护, 从而使整个环境信息模型具有可扩展性、可维护性。共享信息模型由 DTD 和 XML 文档组成, 系统通过对 DTD 的添加、修改、删除以及对 XML 文档的添加、删除等实现对共享信息模型的注册、管理和维护。我们以 (下转第 154 页)

(1) 获取 Plug-in 描述信息和 MIME 类型这两个过程是在 Gecko 启动时进行的, 浏览器会根据提供的信息对插件进行注册。

(2) 初始化资源。浏览器调入 Plug-in 代码时, 调用接口 NP_Initialize, 在这里分配内存和初始化以后各个实例将要使用的资源。初始化过程中, 浏览器为 Plug-in 传递了两个函数指针表: NPNetscapeFuncs 和 NPPluginFuncs。第一个表列举了所有在 Plug-in 里调用的浏览器方的函数地址, 这个表在初始化以前由浏览器填写; 第二个表列举了所有 Plug-in 提供给浏览器调用的函数地址, 在 Plug-in 的初始化过程中填写。在初始化以前所有的 Plug-in 函数调用都是非法的, 因为函数的地址还没有传给浏览器。

(3) 初始化实例。在初始化资源完成以后, 浏览器调用 NPP_New 创建实例, NPP_New 会以指定的 MIME 类型创建实例。在这里定义实例私有的数据。浏览器传递一个特定的结构 NPP 给 Plug-in, 代表实例本身, 此结构体由浏览器创建, 可以将实例私有的数据地址保存于此, 在以后对同一 Plug-in 的函数调用时, 浏览器将这些实例私有数据依次传下去。

(4) 绘制窗口和消息循环。当窗体被创建、移动、改变大小或者销毁时, 浏览器调用 NPP_SetWindow 处理。调用时, 一个含有实例窗口大小、位置和特定系统信息的对象 NPWindow 传给 Plug-in, 在 Plug-in 中根据这些信息绘制窗口。窗口类型分为两类: 有窗体 Plug-in, 浏览器为 Plug-in 创建一个 Motif 窗口部件, 并留出一块专用区域。此类 Plug-in 有自己的消息循环, 可以接收所有来自窗口部件的消息。无窗体 Plug-in, 没有专用区域并且所有消息都是通过浏览器接口 NPP_HandleEvent 传递。

(5) 销毁实例和释放资源。当退出网页、关闭窗口或退出浏览器时, 调用 NPP_Destroy 销毁实例。在这里释放所有初始化实例时所分配的资源。如果有必要也可以通过结构 NPSaveData 保存实例中的信息, 在生成同一 URL 的新实例时浏览器将这些信息原样传入。在所有实例都销毁后, 浏览器调用 NP_Shutdown 释放所有实例共享的资源。

(上接第 151 页) 设计分系统与仿真系统为例说明信息共享的过程: 设计人员通过该平台查询设计系统向仿真系统申请的 DTD 定义, 找到有共享需求的 DTD; 此后, 通过平台提供的“文档另存为”功能由客户端从文件服务器上保存 DTD 和 XML 文档, 或者直接通过 FTP 下载该 DTD 和 XML 文档。设计分系统获取 DTD 和 XML 文档后, 对文档解析, 结合 DTD 从文档内提取所需要的数据, 实现数据共享。提取信息的工作由设计分系统完成, DTD 及 XML 文档的编写在各应用系统明确其提供和需求信息后, 由信息集成小组和用户合作编写。

4 结束语

基于 XML 的信息集成为并行环境下各个分系统之间的信息传递和共享提供了有效的集成手段, 它可以很好地解决企业各个分系统信息的分布性、自治性和异构性。通过信息集成平台, 分系统之间的协同运行得到了实现。本文结合具体的工程

以上阐述了 Linux 下浏览器插件的框架结构和标准接口, 结合本文前面所讨论的视频播放器, 在插件的 NPP_SetWindow 中以管道调用的方式实现视频插件。在网页中的嵌入数据里 MIME 类型指定为 video/x-h263, URL 指定为 rtp: //(端口号), 浏览器会根据这些信息调用以上讨论的视频插件在指定端口接收视频流。

5 结束语

本文在阐明 RTP/RTCP 协议原理的基础上提出了如何实现在一个播放器中播放来自不同源的视频流, 并探讨了在 Linux 下以浏览器插件的方式实现播放。此应用是在原媒体播放器和视频电话基础上进行的重大改进, 而且结合插件技术, 使 Linux 下的多媒体功能更加丰富灵活; 在资源消耗上, 此项改进利用了媒体播放器原有的强大的解码功能, 使视频电话从中脱离出来, 避免了对资源的重复占用。

参考文献:

[1] Schulzrinne, S Casner, R Frederick. RTP: A Transport Protocol for Real-time Applications [S] . RFC 1889 , 1996.

[2] H Schulzrinne. RTP, Profile for Audio and Video Conferences with Minimal Control [S] . RFC 1890 , 1996.

[3] ITU Telecommunication. Standardization Sector of ITU, Video Coding for Low Bit Rate Communication[R] . Draft ITU2T Recommendation. 263 Version 2, 1997.

[4] Netscape Communications. Mozilla Plug-in API Guid [EB/OL] . http: // devedge. netscape. com/library/ manuals /2002/plugin/1. 0 / , 2002-05-10.

[5] H Schulzrinne, R Lanphier. Real-time Streaming Protocol[S] . RFC 2326, 1998.

[6] Equivalence Pty. Ltd. Open H. 323 Library Reference[EB/OL] . http: // www. openh323. org/docs/ OpenH323 / , 2001-10.

作者简介:

蒋勇, 硕士研究生, 研究方向为数据库与管理信息系统; 杜中军, 副教授, 硕士生导师, 研究方向为数据库与管理信息系统; 鞠飞, 硕士研究生, 研究方向为计算机网络。

背景, 采用 XML 进行信息建模, 把统一数据访问平台作为信息集成平台, 实现 CAPE 各个分系统的信息集成。

参考文献:

[1] 谢金崇. 支持并行工程的产品信息集成[D] . 北京: 清华大学(博士后出站报告) , 2001. 12-13.

[2] 胡洁. 多学科并行协同设计理论、方法与应用研究[D] . 北京: 清华大学(博士后出站报告) , 2003. 8-9.

[3] XML 中国论坛. XML 实用进阶教程[M] . 北京: 清华大学出版社, 2001. 37-38.

[4] CAPE 项目初步设计报告[R] . 北京: 清华大学自动化系 CIMS 中心并行工程研究室, 2002. 45-49.

作者简介:

刘科研(1978-) , 男, 河南郑州人, 博士研究生, 研究方向为分布式系统仿真、机电控制; 万丽荣(1965-) , 女, 山东栖霞人, 副教授, 研究方向为并行工程、产品信息集成等; 曾庆良(1965-) , 男, 教授, 研究方向为 CAD/CAM、并行工程、虚拟样机; 范文慧(1966-) , 男, 副教授, 研究方向为并行工程、PDM 产品信息建模。