

企业协同产品研发管理系统^①

韩伟杰¹, 糜旗^{1,2}, 王燕华¹

¹(中国航天科技集团第八研究院 上海航天动力技术研究所, 上海 201109)

²(北京交通大学, 北京 100044)

摘要: 以某企业协同产品研发管理系统为背景, 找出了现有业务中存在的不足, 针对系统中所需要解决的关键问题进行研究. 结合企业实际需求, 对产品全生命周期进行版本控制管理, 基于 J2EE 开发了相应的原型系统, 最后通过实例证明了系统的有效性和可行性.

关键词: 企业协同; 产品全生命周期; 版本控制; J2EE

Enterprise Collaborative Product Development Management System

HAN Wei-Jie¹, MI Qi^{1,2}, WANG Yan-Hua¹

¹(Shanghai Space Propulsion Technology Research Institute, Shanghai 201109, China)

²(Beijing Jiaotong University, Beijing 100044, China)

Abstract: This article is based on one enterprise collaborative product development management system. It finds out the defects among the existing business, and researches on how to solve the key problems in the system. Combined with the actual needs from the enterprise, this article described how to control and manage the version of the whole product lifecycle of product based on the prototype J2EE system. And finally the article proved the feasibility and effectiveness of the system through a sample.

Key words: enterprise collaborative; product lifecycle; version control; J2EE

1 概述

上海某研究所主要从事某军工产品的研制、开发与批量生产, 是具有中、小型固体发动机研发和批生产能力的综合性研究所. 随着科研生产任务的不断增加, 为了满足业务高效有序地开展, 信息化工作面临着巨大的挑战.

企业协同产品研发管理系统 AVIDM(Airspace Vehicle Integration Design & Manufacture)是由北京神舟软件公司开发的一套用于管理企业文档、产品、结构、资料等多功能的大型企业级软件^[1]. 业务部门将该软件用于编制生产计划、下发型号任务、项目计划分解、执行计划任务, 通过对顶层型号产品结构划分, 实现研制计划任务的逐级分解, 通过几年的实施, AVIDM 系统已经在上海某研究所全型号进行了使用.

随着 AVIDM 系统在该场所的不断深入应用, 对

系统本身功能上的需求也在不断增加, 伴随着导弹型号的高密度发射任务, 产品本身面临着研制周期短、生产任务紧的产品研制要求, 从产品研制的实际情况出发, AVIDM 系统自身所存在的一些功能与性能上的缺陷也逐渐显露出来. 由于目前 AVIDM 系统在产品创建过程中仍然采用型号调度手动创建产品的方式, 修改过程具有很大的随意性, 对于产品的基线、版本、结构等全生命周期的把控方面仍未得到有效的控制, 出了问题难以追溯到相应责任人员. 如不能很好的解决这些问题, 将会大大减缓产品工作令下发、试生产的计划进度, 导致产品研制失败.

针对系统中存在的上述问题, 本文针对上海某研究所的业务特点, 以 AVIDM 的二次开发平台为基础, 设计并实现了基于 PDM(Product Data Management)平台的 AVPDM 系统.

① 收稿时间:2015-08-05;收到修改稿时间:2015-10-14

1.1 相关技术综述

1.1.1 基于 PDM 平台的 AVPDM 系统

PDM 平台是以产品结构管理为核心, 将型号研制过程中各阶段产生的各类文档对象、部件对象等, 按照说明关系和参考关系同产品结构结合起来[2,3]. 基于 PDM 平台的 AVPDM 系统采用 Java、workflow 和 Webservice[4-13] 等技术开发, 可与其他工业软件 (AutoCAD、Pro-E) 进行无缝集成.

该平台下的型号产品以航天产品标准规范为基础, 对研制过程中数据可进行查询、借用和修改, 将型号研制生命周期过程中不同类型的数据进行过程管理.



图1 基于 PDM 平台的 AVPDM 总体架构

1.1.2 OSGi

OSGi(Open Services Gateway Initiative)是一个开放式的服务网关协议, 该服务规范由 OSGi 联盟提出, 其体系架构如图2所示[14].

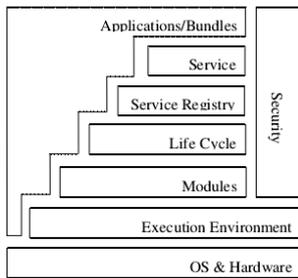


图2 OSGi 结构

OSGi 采用插件式的开发模式(Bundle), 为业界提供了一套通用的、开放的 java 框架[15,16,17], 由于其本身采用的一种是面向服务的编程架构, 使其被广大开发人员所熟知.

1.1.3 多视图 BOM 管理

平台对 BOM 的分类主要分为设计 BOM(EBOM)、工艺 BOM(PBOM)、制造 BOM(MBOM), 在实际的 PDM 系统应用中, 根据不同阶段产品结构管理的要求,

将产生多种类型的 BOM, 从不同角度来表示产品的组成形式以及相关信息.

多视图管理的核心功能包括: 视图版本的管理、视图转换、上下游视图关联以及对等关系维护等.

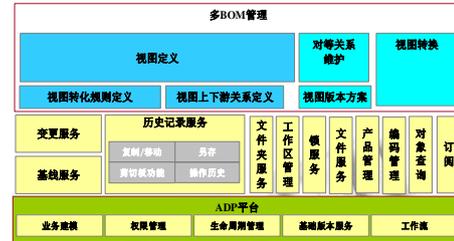


图3 BOM 管理

1.2 版本控制

系统通过版本管理的方式来对产品部件进行控制, 每一个部件对象与三类对象相关: 零部件主对象(Part Master)、大版本(Revision)、和小版本(Iteration). 其中, 零部件主对象 Part Master 上的信息不随版本的变化而变化; 大版本 Revision 代表一个版本系列分支, 大版本升版通过“修订”实现; 同一大版本的各 Iteration 代表了设计的迭代修改, Iteration 的变化通过“检出/检入”实现.

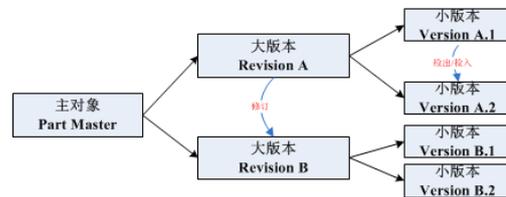


图4 版本控制

从产品分类管理的角度来说, PDM 平台分为部件管理、结构管理、文档管理、上下文管理、生命周期管理. 产品结构依赖的基础服务如图5所示.



图5 产品结构依赖的服务

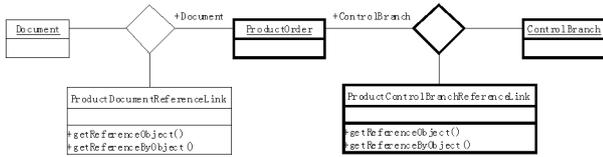


图9 申请单与文档、版本关系类图

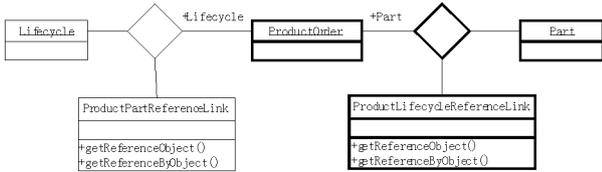


图10 申请单与生命周期、部件关系类图

- ①申请单通过 ProductDocumentReferenceLink 与文档对象建立关联, 得到文档关联对象;
- ②申请单通过 ProductControlBranchReferenceLink 与版本对象建立关联, 得到版本关联对象;
- ③申请单通过 ProductLifecycleReferenceLink 与产品生命周期建立关联, 得到产品生命周期关联对象;
- ④申请单通过 ProductPartReferenceLink 与产品部件建立关联, 得到部件关联对象。

在得到关联对象后, 申请单通过 link 实现系统服务接口类: 产品与文档(ProductDocumentService)、产品与版本(ProductControlBranchService)、产品与生命周期(ProductLifecycleService)、产品与部件(ProductPartService), 从而获取相关对象的信息与服务, 实现方式如下:

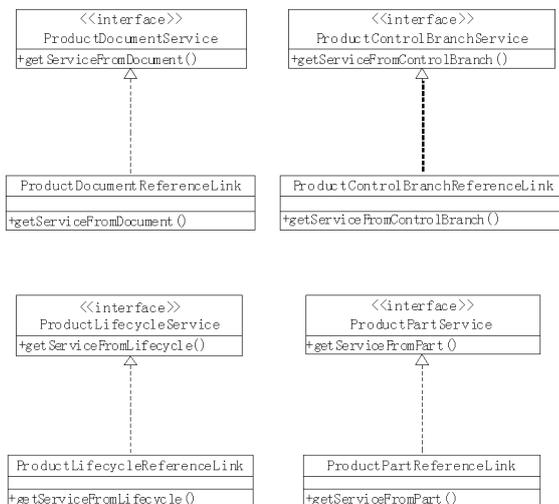


图11 关系 link 与系统接口类图

产品申请管理的服务主要通过与文档、部件、版本与生命周期的使用关系与参考关系、描述关系建立关联, 使用时通过实例化的方式来获取相关服务, 部分代码如下:

```

Part part = PartHelper.getService().newPart();
PartMaster partMaster = (PartMaster)part.getMaster();
    .设置部件主对象属性
    Helper.getIdentifyService().identify(partMaster,
number, name);
partMaster.setEndItem(true);
partMaster.setPropagate(true);
partMaster.setProductId(productId);
    .设置版本对象属性
part.setClassId(classId);
part.setSource(source);
part.setAssemblyPattern(assemblyPattern);
part.setView(view);
    .利用初始化规则设置对象信息
RuleHelper.getService().init(part, context);
Helper.getPersistService().save(part);
.....

```

2.2.2 模块集成与部署

当产品申请模块开发完后, 需要注入PDM平台中依赖的相关扩展点, 在 plugin.xml 中写入相关属性对象 extension:

```

<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.4"?>
    <plugin id="plm.plugin.003.productApply"
autostart="true">
        <extension
            point="adp.plugin.web.jsp">
        </extension>
        <extension
            point="adp.plugin.web.webresource">
            <resource
                alias="/productApply/webapp"
                dir="/webapp">
            </resource>
        </extension>
    </extension>

```

```

point="adp.plugin.orm.orm">
</extension>
<extension
point="plm.plugin.grid.spot">
<spot
id="PRO.ROLE.MEMBER">
<types>
<type
id="User"
view="true">
</type>
<type
id="Role"
view="true">
</type></types>

```

① 在系统部署时首先先进入 OSGI 插件控制台，确保平台中 bundle 插件的完整性；



图 12 插件控制台

② 将产品申请模块插件部署到平台，开启 orm、springframework、ui 等扩展点插件，确保插件正常运行。

ID	Name	Version	Category	Status	Actions
0	System Bundle (org.apache.felix.framework)	4.2.0		Active	
1	Apache SisuPlexor (org.apache.geronimo.sisu.plexor)	1.0.2.RHEL6.A026		Active	
2	ADP Alliance API (com.gangwonitae.org.alliance)	1.0.0		Active	
3	OSGIDevTool (org.apache.felix.devtool)	3.0.0		Active	
4	Apache Felix Web Console (org.apache.felix.webconsole)	3.2.1		Active	
5	Apache SisuPlexor-Extender (org.apache.geronimo.sisu.plexor-extender)	1.0.2.RHEL6.A026		Active	
6	Spring Core (org.springframework.core)	3.0.0.RHEL6.A026		Active	
7	OSGIDevTool - Dev Support (org.osgidevtool.dev-support)	3.0.0		Active	
8	Spring AOP (org.springframework.aop)	3.0.0.RHEL6.A026		Active	
9	Apache Felix Framework (org.apache.felix.framework)	4.2.0		Active	
10	Common Logging (org.apache.commons.logging)	1.1.0		Active	
11	Apache Felix Web Console (org.apache.felix.webconsole)	3.2.1		Active	
12	Spring Context (org.springframework.context)	3.0.0.RHEL6.A026		Active	
13	Spring Beans (org.springframework.beans)	3.0.0.RHEL6.A026		Active	
14	Spring Expression Language (org.springframework.expression)	3.0.0.RHEL6.A026		Active	
15	Apache Felix Web Management Console (All In One) (org.apache.felix.webconsole)	4.2.0.A01		Active	
16	Spring AOP (org.springframework.aop)	3.0.0.RHEL6.A026		Active	

图 13 插件部署

2.2.3 系统实现

产品申请模块功能设计如下：

① 登录 AVIDM 系统，通过点击“产品申请单”菜单，在右侧页面中点击“创建”按钮，弹出“产品申请单”创建页面，在创建页面中，由创建人填写型号代号、型号名称、型号区分代号、产品代号、产品名称，下拉选择“所属标签”属性值。



图 14 产品申请单管理页面

图 15 创建产品申请单

② 在“产品申请单创建”页面中，点击“阶段序列”，弹出“阶段序列信息”页面。进入页面后，创建人可以添加阶段，并可以为初始阶段添加子阶段；移动阶段时先选中需要移动的目标阶段，可对阶段进行移动；对阶段也可以进行移除操作，所有信息设置完成后，所操作的信息会返回到“阶段序列页面”中显示。



图 16 阶段序列页面

③ 在“产品申请单创建”页面中，创建人点击“团队”按钮，弹出“团队页面”，创建人选择“角色及成员”后，可以添加角色，并为角色添加用户、组、组织机构，所有信息设置完成后，所操作的信息会返回到“角色和成员页面”中显示。另外，创建人也可以选择“专业及人员”，添加专业，并为专业添加参与人员和参考人。

员,所有信息设置完成后,所操作的信息会返回到“专业页面”中显示。



图 17 角色和成员页面



图 18 专业及人员页面

④ 创建人填写完“产品申请单”相关信息后,点击“操作”按钮,选择“送审”,系统自动弹出选择 workflow 模板页面,创建人选择“产品申请单”对应的工作流模板,指定各审批环节的审批人员,指定完毕后,点击“确定”。

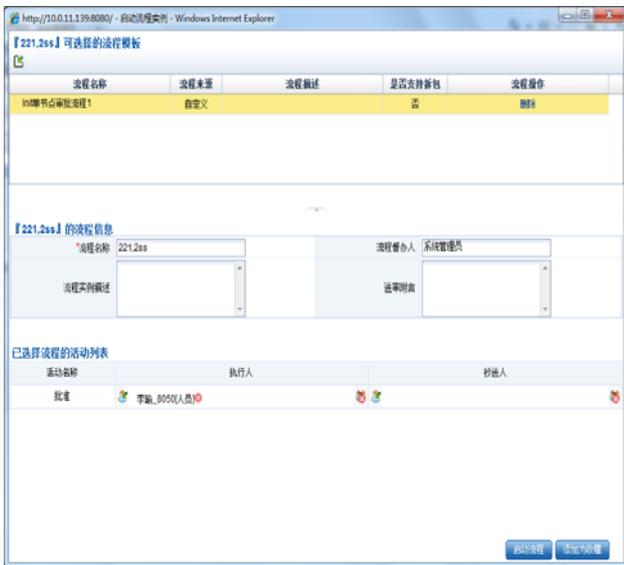


图 19 申请单送审页面

⑤ 用户点击“产品申请单管理”菜单,在右侧的“产品申请单”列表中点击某条“产品申请单”信息,进入其详细信息页面,在“详细信息”选项卡中查看“产品申请单”的详细信息:包括常规属性信息、系统属性信息、业务属性信息等。



图 20 详细信息页面

⑥ “产品申请单”审批完成后,新一代 AVIDM 系统根据“产品申请单”相关信息自动创建产品。



图 21 产品信息页面

整个产品的生命周期流程“设计中—审批中—审批通过—受控”,当产品申请单通过相关人员审批后,该产品创建成功,初始产品版本为 A.1(系统版本规则)。

3 结语

随着 AVIDM 系统在某研究所内的不断应用,以信息化手段为核心,从根本上改变了原有企业文档、产品的设计管理模式,缩短了产品研制过程的生命周期。由于系统本身采用插件式的二次开发模式,使得平台本身具有易维护、可扩展的特点,日后为企业从设计、生产、制造设计一体化提供基础保障。

参考文献

- 1 北京神舟航天软件技术有限公司.神舟 AVIDM5.0 产品结构技术说明,2014.
- 2 徐宗云,王伟伟.基于 SOA 架构的 PDM 系统与制造管理系统集成设计与实现.冶金自动化,2010,34(1):10-15.
- 3 王文理,李海英,等.某设计制造一体化应用工程中的集成制造分系统的设计与开发.航空制造技术,2004,8:73-76.
- 4 Sun FQ, Wang L, Chen T, Qu YL. Dynamic information integration of virtual enterprises based on web services and J2EE. International Conference on Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. 2007. 6146-6149.
- 5 Curbera F, Duftler M, Khalaf R, et al. Unraveling the Web services web: An introduction to SOAP, WSDL, and UDDI,

- Internet Computing, IEEE, 2002, 6(2): 86–93
- 6 Georgakopoulos D, Hornick MF, Sheth AP. An overview of workflow management: From process modeling to workflow. Automation Infrastructure, 1995(2).
- 7 Reijers HA. Design and Control of Workflow Processes, Business Process. Artech House, 2006.
- 8 晏婧. workflow 系统开发图解 workflow 的基本路由方式. 计算机应用, 2010, 11: 15–18.
- 9 Huang JS, Gutierrez RLZ, Garcia BM, et al. Sharing ontology schema information for web service integration. Proc. of the Fifth International Conference on Computer and Information Technology. 2005. 421–442.
- 10 曲海成, 刘万军. 基于 Struts-Spring-Hibernate 框架的软件复用技术. 计算机与现代化, 2008, (6).
- 11 袁嘉. 基于 Struts+Spring+Hibernate 的 Web 开发框架. 网络安全技术与应用, 2008, (4).
- 12 Gueheneuc YG, Antoniol G. DeMIMA, a multilayered approach for design pattern identification. IEEE Trans. on Software Engineering, 2008, 34(5): 667–684.
- 13 Gamma E, Helm R, Johnson R, et al. 设计模式: 可复用面向对象软件的基础. 北京: 机械工业出版社, 2000.
- 14 OSGi Alliance. RFC 119 Specification, <http://www.osgi.org/Specifications/HomePage/>, 2009.
- 15 Oscar. An open source OSGi implementation. <http://oscar-OSGi.sourceforge.net>.
- 16 OSGi Alliance. OSGi Service Platform Version 4.2 Specification, <http://www.osgi.org/Specifications/HomePage/>, 2010.
- 17 OSGi Alliance. OSGi Technical Whitepaper, <http://www.osgi.org/Links/HomePage/>, 2007.