

跨站点协同流程研究与应用

北京航天长征飞行器研究所 赵振杰 张明星 王林林
北京航天动力研究所 李成龙

针对流程在多个站点间无法进行协同审签的问题，提出了基于 Teamcenter 系统多站点协同功能及二次开发技术的跨站点协同流程解决方案，阐述了系统功能的实现原理及实现过程并在产品设计文件审签流程中开展应用。

一、引言

西方工业研制模式经过几百年不断发展与完善，产品设计、工艺和制造已融为一体，形成从设计、生产、交付、运维至报废的一个完整的产品研制链路。而中国的工业研制模式自建国以来一直以效仿苏联为主，形成了设计单位和生产企业相分离的独具特色的“中国工业体系二元制”特征。实际上，在产品研制的全生命周期中，无论是设计、工艺、仿真、生产制造还是运维，都应该是一体化的，在当前数字化设计与制造的趋势下，这种矛盾显的更加突出。

航天产品自研制以来，一直遵循设计所和生产单位相分离的模式，由于航天产品的研制流程涉及设计、工艺等多家研制单位，传统的二维产品研制流程需要设计与工艺人员在图纸上共同签字才能生效。随着产品研制模式由二维设计与工艺串行研制模式向三维设计与工艺并行协同模式转变过

程中，由于设计单位与制造单位用于产品技术状态管理的 PLM (Product Lifecycle Management, 产品全生命周期管理) 系统独立部署，同一个产品研制流程无法在两个 PLM 系统进行流转。目前普遍采用的解决方法有两种：一种是工艺人员登录设计 PLM 系统进行审签，一种是工艺人员在纸质会签跟踪卡上进行签字，并由设计人员在设计 PLM 系统进行代签。在航天产品协同研制日益普及的背景下，上述两种办法显然无法满足跨域协同流程的审签需求。

西门子公司的 Teamcenter 作为一个企业级的商业应用 PLM 软件，提供了良好的开发接口，具有极大的开放性，用户可以方便地对其服务端及客户端进行二次开发，满足用户的使用需求。本文通过 Teamcenter 多站点功能和服务端、客户端的二次开发技术实现产品设计文件流程在设计所与生产单位独立的 Teamcenter 系统中进行审签，达到流程跨站点协同的目标。

二、基本原理

作为全球知名的 Teamcenter 软件是一个体现协同应用、行业解决方案以及具有产品全生命管理等诸多功能于一体的

PLM 平台，在流程协同应用方面也提供了基本解决方案，即 Teamcenter 通过“远程任务箱”功能处理远程站点发送的流程任务，通过分配相应的远程任务查看与操作权限，Teamcenter 允许在生产单位站点对设计所远程站点发送过来的流程任务进行处理。用户在接收到远程站点发送过来的流程任务时，通过双击远程任务箱登录远程站点的四层瘦客户端进行审签，完成流程审签任务。但由于生产单位人员进入设计所站点进行流程审签的前提是在设计所站点创建工艺人员账户，导致设计所 Teamcenter 系统大量的登录账户被工艺人员占用，在流程协同时此种解决方案实际并没有被认可。因此本文提出利用 Teamcenter 多站点结合二次开发技术的解决方案。

下面以航天产品设计文件审签流程为例，说明跨站点协同流程的实现原理，设计文件审签流程如图 1 所示。



图 1 设计文件审签流程

上述流程为航天产品设计文件的标准审签流程，其中工艺主管和工艺会签是由生产单位工艺人员进行审签。整个协同会签流程由设计所设计人员发起，并指派各个节点的审签人员，包括工艺主管和工艺会签人员，但工艺主管和工艺会签人员进行指派时须选择通过程序解析出来的生产单位的虚拟组织人员。

流程执行审签过程时，设计、校对、审核、标审及批准为系统标准审签流程，按 Teamcenter 系统正常审签流程执行即可，当流程执行到“工艺主管”时，程序自动在生产单位的 Teamcenter 系统自动发起跨站点审签流程，此流程只包含“工艺主管”和“工艺会签”两个节点。

当生产单位的工艺主管及工艺会签人员进入 Teamcenter 系统审签跨站点流程时，通过二次开发程序，自动将审签的结果反馈到设计所 Teamcenter 系统中对应的流程中，设计所 Teamcenter 系统在接收到流程反馈结果后，通过后台的二次开发程序，自动执行“批准”或“拒绝”操作。

当在生产单位的 Teamcenter 系统中用户执行“拒绝”操作时，系统通过二次开发程序自动删除当前工作流程，同时设计所的 Teamcenter 系统在接收到操作结果后自动将流程驳回到设计员节点。设计员依据修改意见，完成设计文件的修改后再次提交流程进行审签。

三 . 关键技术

1. Teamcenter 二次开发关键技术

Teamcenter 二次开发分为客户端和服务端开发，客户端开发采用 JAVA 语言，利用 Eclipse 工具进行开发；服务器开发采用 C 语言及集成工具包（Intergrated ToolKit，简称 ITK），利用 Visual Studio 工具进行开发。

(1) 客户端开发。

客户端开发主要用于在 Teamcenter 软件中扩展菜单、视图和透视图等，依据客户需求定制一些对话框、操作界面或功能模块等，通过功能扩展可以使系统更好地服务于实际业务，在跨站点协同流程中客户端开发主要涉及扩展菜单功能，添加菜单主要分 3 个部分：command：每个定制菜单所需的唯一识别 ID 号，代表一个操作的抽象意义，如打开、退出等。Handlers：通常继承于 AbstractHandler 类，它代表菜单中执行进实际进行的具体动作，通过它执行菜单工具条所需表示的命令。Menu Contributions：代表菜单出现的视图位置，通过它创建菜单和工具条的接口框架并且控制显示位置。客户端菜单开发代码如下。

```

<?xml version="1.0" encoding="UTF-8"?>
<plugin>
  <extension point="org.eclipse.ui.commands">
    <command
      name="Command"
      id="com.addmenuitem.commands.Command">
    </command>
  </extension>
  <extension point="org.eclipse.ui.handlers">
    <handler
      commandId="com.addmenuitem.commands.Command"
      class="com.addmenuitem.handlers.Handler">
    </handler>
  </extension>
  <extension point="org.eclipse.ui.menus">
    <menuContribution locationURI="menu:tools?after=ad
ditions">
      <command
        commandId="com.addmenuitem.commands.Command"
        icon="icons/sample.gif"
        id="com.addmenuitem.menus.Command">
      <visibleWhen>

```

```

<reference
    definitionId="com.teamcenter.rac.
ui.inMainPerspective">
    </reference>
</visibleWhen>
</command>
</menuContribution>
</extension>
</plugin>

```

(2) 服务端开发。

服务器端的全部功能模块是建立在一组集成开发工具 (Intergrated ToolKit, ITK) 之上的应用模块, 服务端开发由应用层和核心层构成。

应用层包括对象接口组件适配器 (Object I/F Comp Adapters)、应用对象 (Application Objects) 和应用 (Applications), 其中对象接口组件适配器通过 CORBA/DCOM 等技术实现与 CAX 工具等其他外部应用程序的通讯, 从而实现系统与外部程序的集成, 应用程序包括 Teamcenter 中的应用模块, 如我的 Teamcenter、产品结构管理器、分类等, 它们使得客户层能够浏览和访问服务层的数据信息。

核心层由 POM (Persistent Object Manager) 管理器和数据库 I/O 模块 (Database I/O Module) 组成, POM 管理的对象是持久对象即存储在数据库的对象, 持久对象管理器连接应用工具模块和系统模块, 解析应用工具层命令, 并产生相应的 SQL 语句, 在数据库中产生相应的关系数据, 使得应用层能操作数据库。数据库 I/O 模块提供数据库操作接口, 实现对数据库的操作。

2. Active MQ 服务

Active MQ 是 Apache 公司出品一款最流行的、能力强劲的开源总线, Active MQ 是一个完全支持 JMS1.1 和 J2EE1.4 规范的 JMS Provider 实现。它非常快速, 支持多种语言的客户端和协议, 而且可以非常容易的嵌入到企业的应用环境中, 为应用程序提供高效的、可扩展的、稳定的和安全的的企业级消息通信。

3. 多站点协同

可用于在多个 Teamcenter 数据库之间交换数据对象, 数据库间的对象传递由指定服务器上运行的守护程序进行控制, 对象的复制是通过将对象从原来数据库中导出, 再将它们导入请求数据库来进行。

四. 系统功能实现

跨站点协同流程实现过程主要包含两部分内容: 跨站点用户组织管理和跨站点审签流程管理。

1. 跨站点用户组织管理

通过 Teamcenter 系统的服务端端的二次开发及配置, 实现跨站点组织机构自动导出、数据推送、文件接收及组织信息的解析, 执行过程如图 2 所示。

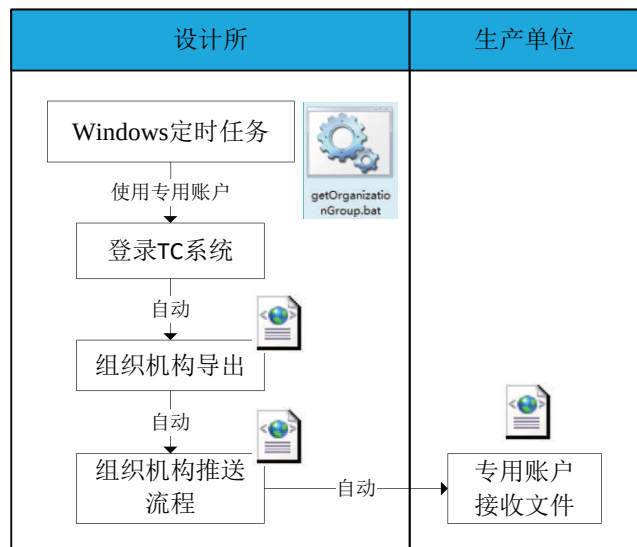


图 2 跨站点用户组织管理

通过 Windows 定时任务自动调用二次开发程序, 生产单位 Teamcenter 定期将组织机构

信息 (包括组、角色、用户及逻辑关系) 以 XML 格式导出, 如图 3 所示, 再通过多站点协同功能自动推送到设计所 Teamcenter 系统。设计人员在发起设计文件审签流程时, 在工艺主管及工艺会签节点选择人员时程序自动解析生产单位发送过来的包含组织机构信息的 XML 文件, 在流程人员选择界面以树状结构显示, 形成外部站点的虚拟组织, 如图 4 所示。

```

<?xml version="1.0" encoding="utf-8" ?>
<PLMXML>
  <Info sequence="1" name="site2" type="组" />
  <Info sequence="1.1" name="site2role" type="角色" />
  <Info sequence="1.1.1" name="user21 (user21)" type="用户" />
  <Info sequence="1.1.2" name="user22 (user22)" type="用户" />
  <Info sequence="1.1.3" name="user23 (user23)" type="用户" />
  <Info sequence="2" name="site2role1" type="角色" />
  <Info sequence="2.1.1" name="user31 (user31)" type="用户" />
  <Info sequence="2.1.2" name="user32 (user32)" type="用户" />
  <Info sequence="2.1.3" name="user33 (user33)" type="用户" />
  ....
</PLMXML>

```

图 3 组织机构信息文件



图 4 外部站点虚拟组织

2. 跨站点审签流程

(1) 子流程发起。

在进行跨站点协同流程时，设计所在系统发起设计文件审签主流程，当流程执行到工艺主管节点时，通过 Teamcenter 系统的多站点协同功能自动将流程目标下的对象发送到生产单位站点，同时，程序通过 Active MQ 服务将当前流程任务的目标对象、当前节点的用户信息、设计所站点信息发送至生产单位站点中，生产单位站点调用二次开发程序，实现子流程自动发起，并自动指派节点审核人员，如图 5 所示。

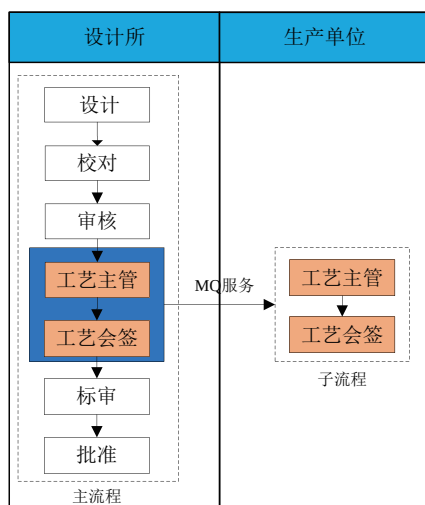


图 5 协同流程子流程

(2) 子流程审签。

首先在生产单位的 Teamcenter 系统工艺协同子流程中添加由 ITK 开发的 handler 程序，如图 6 所示，用来实现在

子流程审签执行完毕后，程序自动调用 Active MQ 服务，将子流程的审签结果反馈至设计所。

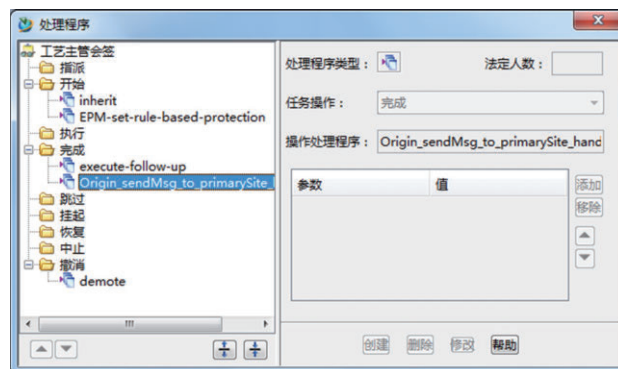


图 6 添加 handler 程序

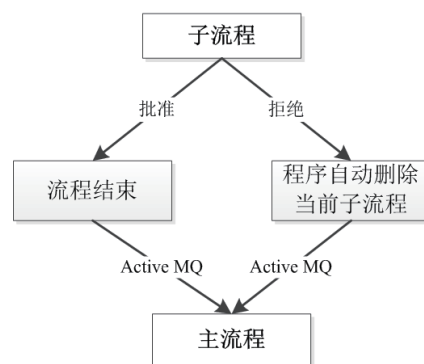


图 7 子流程审签

子流程审签时，如果工艺人员执行“批准”操作，则当前流程结束且程序自动调用 Active MQ 服务将审签信息发送至设计所站点；如果工艺人员执行“拒绝”操作，则程序调用 Active MQ 服务将审签信息发送至设计所站点后自动从系统中删除子流程，子流程审签如图 7 所示。

当设计所收到子流程发送过来的审签信息后，程序根据审签结果，自动提升或回退主流程审签节点。

(3) 日志记录。

通过服务端开发实现设计文件跨站点审签流程实际是将同一个流程分为两个流程并在两个站点的 Teamcenter 系统内执行，由于 Teamcenter 系统的审签日志只能记录在流程所有者站点中，因此协同子流程的审签信息无法在设计所站点中进行完整记录。

通过对 Teamcenter 系统客户端进行二次开发，实现真正意义上完整的审签日志记录，程序自动将设计所主流程的审签信息与生产单位子流程的审签信息统一在设计所

Teamcenter 系统数据库表中进行记录，其实现代码如下。

```
public class ReviewInformationAccessCommand extends
AbstractAIFCommand{
    private AbstractAIFUIApplication app;
    private TCSession session;
    private InterfaceAIFComponent[] targets;
    .....
    public ReviewInformationAccessCommand(AbstractAI
FUIApplication abstractaifuapplication){
        this.app = abstractaifuapplication;
        this.session = ((TCSession)this.app.getSession());
        this.targets = this.app.getTargetComponents();
        if (this.targets[0] instanceof
TCComponentItemRevision){
            try {
                TCComponentItemRevision itemRevision =
(TCComponentItemRevision)this.targets[0];
                AIFComponentContext[] coms =
itemRevision.whereReferenced();
                for (int i = 0; i < coms.length; i++)
                    if ((coms[i].getComponent() instanceof
TCComponentTask)) {
                        task = (TCComponentTask)coms[i].
getComponent();
                        job = (TCComponent)task.
whereReferenced()[0].getComponent();
                        task_id = job.
getTCProperty("root_task").getReferenceValue().
getProperty("comments");
                        break;
                    }
            }
        }
        .....
    }
}
```

五. 应用案例

航天某设计所与其生产单位分别部署独立的 Teamcenter 系统，并且通过 ODS、IDSMS 服务实现两个站点互联互通及数据相互发送，但由于产品设计文件流程无法在两个站点间流转进行审签，导致工艺人员需要进入设计所 Teamcenter 进行会签，增加了设计所 Teamcenter 系统应用成本及管理难度。通过在两家单位部署跨站点协同流程审签功能，在不改变设计及工艺人员审签应用模式的前提下，实现了设计人员及工艺人员分别在本单位的 Teamcenter 系统进行设计文件的审签，解决了设计所 Teamcenter 系统大量

license 被工艺人员占用的问题，工艺人员进行设计文件审签如图 8 所示。

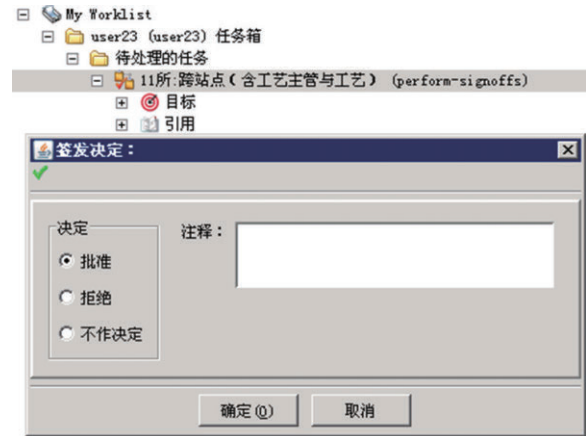


图 8 工艺人员审签界面

设计文件审签流程审计签完毕后，设计人员通过二次开发菜单查阅流程完整的审签信息，如图 9 所示。

任务名称	节点名称	用户信息	转发后用户信息	日期	审签意见	注释
000138/A-连接板	设计	liuhj (刘洪杰)		2017-10-11 09:15	完成	
000138/A-连接板	校对	liuf (刘峰)		2017-10-11 09:48	批准	
000138/A-连接板	审核	jiangq (姜绪强)		2017-10-11 13:56	批准	
000138/A-连接板	外部会签	zhanglf (张龙飞)		2017-10-12 09:36	批准	
000138/A-连接板	外部会签	lizhd (李振东)		2017-10-12 11:10	批准	
000138/A-连接板	标准	yueht (岳昊瞳)		2017-10-13 14:03	批准	
000138/A-连接板	批准	chubx (褚宝鑫)		2017-10-13 16:56	批准	

图 9 协同流程审签信息

六. 结语

本文对跨站点协同流程的相关问题进行分析和研究，通过 Teamcenter 系统的二次开发技术，完成了跨站点协同流程的开发，此功能已成功应用于航天某型号产品设计文件审签流程中，满足了厂所协同研制过程中流程跨站点应用的需求，提高了跨站点流程审签效率。IM