# 基于工作流的软件过程开发平台的研究*

邹晓宇, 王　戟, 周俊鹏
(国防科学技术大学 计算机学院, 湖南 长沙 410073)

**摘　要**: 首先介绍了传统软件过程中人为因素的影响,并给出了减少这种影响的解决方法:在工作流与软件过程相结合的基础上集成软件过程开发平台 SPDET( Software Process Develop Environment Tool) 。然后详细的讨论了 SPDET 的几个重要的组成部分,并给出了 SPDET 的设计和实现。

**关键词**: 软件过程; 工作流; 过程建模
**中图法分类号**: TP31　　**文献标识码**: A　　**文章编号**: 1001-3695( 2006) 01-0043-04

## Research on Workflow-based Software Process Development

ZOU Xiao-yu, WANG Ji, ZHOU Jun-peng
*( School of Computer Science, National University of Defense Technology, Changsha Hunan* 410073, *China)*

**Abstract:** At first, the paper introduces the effect of the human factors in the traditional software process, and proposes a solution of this problem. The solution is to build the SPDET ( Software Process Develop Environment Tool) based on the workflow system. And then discusses the principles for building the SPDET, and describes its design and implementation.
**Key words:** Software Process; Workflow; Process Modeling

"软件危机"的出现使得软件工程成了软件研究的中心,探索新的软件开发方法和技术以提高软件的质量和生产率一直是软件工程学的一个焦点。软件工程也取得了显著的成就,一些项目在软件工程的指导下取得了成功,但是软件危机并没有得到很好的解决,大规模的软件生产仍然得不到很好的控制。M. Dowson 在 IEEE 会议上指出: "软件产品的质量在很大程度上依赖于软件过程,尤其是大规模的软件开发更是如此"。于是从 20 世纪 80 年代开始,以研究软件过程为内容的软件过程技术成为控制和管理软件质量的研究热点。

随着研究的深入,发现在软件过程中人为管理因素的影响而造成的软件开发失败的事例占了很大的比例,一些项目虽然在开始制定了周详的项目开发计划,但是很少完全按照计划来执行,所以软件界希望能用一种相对自动化的过程工具来代替人的大部分工作,进而减少人为因素的影响。于是人们把研究的目标转向了开发出一种可以支持软件过程的管理工具上来,而且不断地有新的过程管理环境的出现。据 David Sharon 和 Rodney Bell 的统计,近年来推出的过程管理环境主要包括以下几方面的功能:①建模,用于项目建模并将之映射到工具支持的任务上;②工具协调,用于将工具的使用协调起来以便于支持过程中的各个任务;③运作,用于指导和支持项目的有关人员遵循开发过程进行各项活动;④管理,用于帮助项目管理人员确定开发活动对过程完成的影响,根据实际情况作出相应的决策。这些工具的代表有:①InConcert;②Process Weaver;③SynerVision;④Process Engineer;⑤SE Companion 等。

过程环境工具是一个集成的过程开发和运作环境,它描述了工具与环境中其他元素的关系,这些元素包括工具、平台和过程,而工具之间的关系以及这些关系的特性则是集成的关键,工具之间要有一致的范围,包括数据格式、用户界面和公共函数的使用以及工具结构等。与现有的过程环境相比,本系统是根据软件过程的基本特性并结合现今已经比较成熟的工作流技术而得到的集管理和开发一体的过程环境。它是通过软件过程建模语言对软件开发过程建模而得到的适合工作流的过程模型,工作流引擎解析前面定制的过程模型,并按照解析的结果自动或半自动地执行软件过程的开发流程。

基于上述策略,深入研究了工作流在软件开发过程中的应用技术,设计并实现了一个基于工作流的软件过程应用开发平台 SPDET。

## 1　基于工作流的软件过程

### 1.1　软件过程

自 1967 年 NATO 研究组织首次提出了"软件工程"这一术语以来,软件工程的理论和方法得到了极大的发展。目前软件过程作为软件的业务过程已经成为软件开发人员在工程领域的重要研究方向之一,普遍的观点认为软件产品的质量跟软件过程的质量有密切的关系。在第九届国际软件工程会议上,L. J. Osterweil 提出了"软件过程仍然是软件"的著名论断[1],为软件过程的研究和发展提供了理论依据。与其他企业的生产过程相比,软件过程具有复杂的特性:

①软件开发离不开人的参与,而且作为参与者的人的脑力劳动是软件活动的主要活动之一;

②软件开发人员之间的协作与交流在软件开发中不可或缺,有时甚至是软件成败的关键因素;

③软件过程本身也是需要在软件开发实施过程中得到不断的完善。

收稿日期: 2005-01-21; 修返日期: 2005-03-10
基金项目: 国家"863"计划资助项目( 2003AA001023)

针对以上特点,在第一届 ISPW 研讨会上正式提出了软件过程的概念和明确的定义:软件过程是在软件生存周期中所实施的一系列活动的集合,并且每个活动都是由一系列的任务组成的。随着对软件过程研究的深入,人们认识到软件过程是软件生存周期中一系列的相关过程,过程就是活动的集合,活动就是任务的集合,任务的作用就是将输入转换为输出,并且活动可以顺序地、迭代地、并行地、嵌套地和有条件的触发。

### 1.2 工作流管理系统

工作流管理联盟(Workflow Management Coalition, WfMC)给工作流的定义是:工作流所要解决的主要问题是使在多个参与者之间按照某种预定义的规则传递文档、信息或任务的过程自动进行,从而实现某个预期的业务目标,或者是促使此目标的实现。在 WfMC 及其广大工作流工作者的努力下,工作流技术已经是一项成熟的技术,实践证明工作流管理技术已成功运用到图书馆、医院、保险公司、银行、工业制造等行业。

工作流管理系统(Workflow Management System, WfMS)是定义、创建和执行工作流的系统,也是以计算机支持的分布式、协同工作业务流程的自动或半自动化为研究目标的软件系统。不同工作流管理系统可以有不同的实现方法,不同的底层通信机制,应用的范围也可能有很大的差距,但所有的工作流管理系统从用户的应用层上来看,通用工作流管理系统应该能够提供以下三个方面的功能支持:

(1) 建造功能。对工作流的业务流程及其组成的活动进行定义和建模。

(2) 运行期控制功能。即在一定的运行环境下,负责创建、执行和控制工作流实例,激活相应的资源和应用,并完成过程中从一个活动到另一个活动的控制转移。它是整个工作流管理系统的核心部分。

(3) 运行期交互功能。在动态运行过程中,WfMS 与用户(业务的参与者或控制者)以及外部应用程序工具进行交互。

### 1.3 软件过程与工作流的结合

在软件工程的发展过程中人们不断的总结软件工程的经验教训,从中得到了许多新的软件开发模型,比如从最原始的瀑布模型到快速原型模型,然后是 RUP 的迭代模型等。这些模型的出现都是为了解决日益复杂的软件开发过程。要想使软件过程与工作流有机地结合起来,必须在软件过程建模的时候考虑到过程复杂性与工作流程的易控制性的平衡点,因而不能完全采用迭代模型,而是采用瀑布模型与迭代模型相结合的策略。根据以上策略抽象出一个简单的软件开发过程模型,在该模型中,把软件开发流程分为六个活动:需求分析、概要设计、详细设计、编码、测试、部署。在该流程中,每一个活动都由一个特定的活动参与者来执行。在该流程模型中共有六个活动参与者,分别为:需求分析师、概要设计师、详细设计师、程序员、测试工程师、项目经理。需求分析师、概要设计师、详细设计师、程序员、测试工程师五个参与者执行完活动之后,软件过程中相关的产品都提交到项目经理,由项目经理对他们已完成的任务进行审批。

基于软件过程的工作流是一种特殊的工作流管理系统,因为软件过程的活动离不开参与者创造性劳动,不同的人之间不仅仅需要文档的传递,还需要人之间的交流。正是这些人为的因素导致了流程并不是完全的按顺序执行。为了解决这个问题,可以简单地设定大活动之间是按顺序执行的,而每个活动内部的流程可以根据情况迭代式的执行。每个活动在最后通过项目评审后表示该活动已经结束,并形成相应的文档放在特定的目录。

## 2 软件过程开发平台 SPDET 的设计

### 2.1 SPDET 设计原则

由于考虑到可视化的图形界面和 Client 要求被动地接受任务消息,所以系统采用 Client/Server 模式。系统分为客户层、中间层和数据层三层,其中最上层是提供图形用户界面的客户层,用户可以借助于方便的界面元素来与系统交互。在客户层,用户可向应用服务器(中间层)请求服务,并且接收服务器返回来的结果,显示在用户界面上;中间层是一个应用服务器,服务器里面部署了一些负责处理业务逻辑的应用组件:引擎(其中包括了消息管理器和其他功能部件);底层是一个关系型数据库,提供持久的存储机制。

三层构架的优点:把复杂的业务逻辑的处理放在了中间层来处理,客户端进行简单的业务处理和显示工作,这样的话系统对客户端的硬件需求不高,客户端的维护也相对来说变动不大,在业务逻辑发生变化的时候只需要对服务器上的中间层进行维护就可以了。这样的系统结构在分布式应用中更能体现出优越性。系统设计逻辑图如图 1 所示。

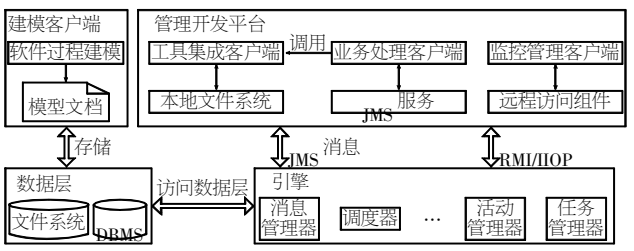

图 1 系统设计逻辑图

### 2.2 客户层

客户层是一个与用户进行交互的接口。根据系统需要分为业务处理客户端、监控客户端、工具集成客户端和建模客户端四个部分。

#### 2.2.1 业务处理客户端

客户端主要实现三个方面的功能:接受任务、处理任务和提交任务,在处理任务的时候就需要调用系统集成的开发工具来完成相关的任务。接受任务是指角色第一次查看引擎派发的任务,同时系统自动地把该任务写入数据库(DBMS),并把任务状态设置为已读。此时的任务在 JMS 的主题里面删除掉了,角色没有完成并提交任务之前只能从数据库里面取得已经接受的任务。处理任务是指用户调用集成的工具进行业务处理,工具的信息保存在文件系统里面以供其他的工具调用。提交任务是当角色完成了具体的任务时,把完成的文档提交给引擎,由引擎统一存放在文件系统里面,同时删除数据库里面的任务信息。如果任务评审不通过,则引擎根据流程再次把任务派发给角色,同时还会把相关的具体业务处理信息派发给角色。功能需求图如图 2 所示。

为了实时地实现用户任务的接收,业务处理客户端采用发布/订阅(Pub/Sub)模型异步消息的模式。系统采用 WebLogic 的 JMS 消息服务,实现是在应用服务器(WebLogic)上部署一

个消息管理器,它是引擎的一个通信组件负责引擎与外界的消息通信,同时在本客户端也实现了一个 JMS 消息收发器,可以异步地从服务端接收实时的引擎消息或者发送消息给引擎。

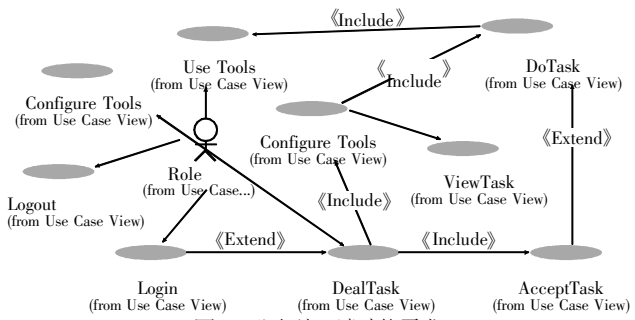


图 2　业务处理端功能需求

客户端流程如图 3 所示: 首先是角色登录,登录后系统根据用户的类型来确定进入的用户界面,一般用户进入应用开发界面,而管理员用户则进入监控管理界面。一般用户可以浏览引擎发送过来的任务,通过浏览任务信息来了解自己接收的任务,接下来即是根据任务描述来完成相关的任务。对于完成任务需要的相关资料比如说文档模板、前期开发中的需求分析文档或者设计文档等,采用附件的形式跟随任务消息发送给角色。完成任务后用户可以把自己工作中的文档以附件的形式上传,引擎接收到角色提交的任务后,把它发送给项目负责人来进行评审,评审的结果以文档的形式保存在文件系统中的特定目录以供引擎读取。

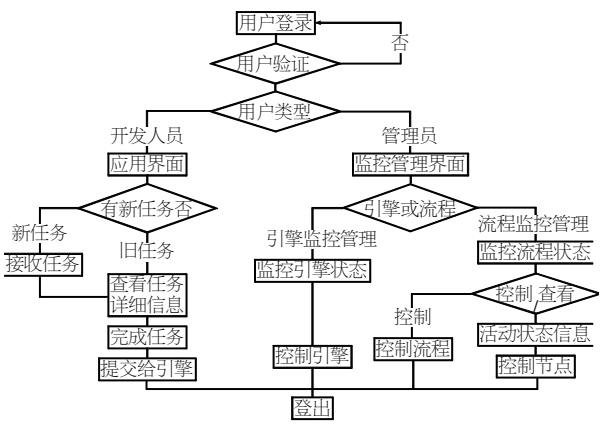


图 3　应用、监控客户端流程图

### 2.2.2　监控管理客户端

监控管理客户端的主要功能是为引擎提供了一个监控和管理的接口,在设计阶段,监控管理客户端可以帮助工作流设计者调试工作流模板,定位其中的错误设计和结构; 在运行阶段,系统监控客户端可帮助系统管理员监控和干预工作流实例的执行。其主要功能是对引擎及其引擎中的流程进行实时监控,并提供管理的功能。其功能需求图如图 4 所示。

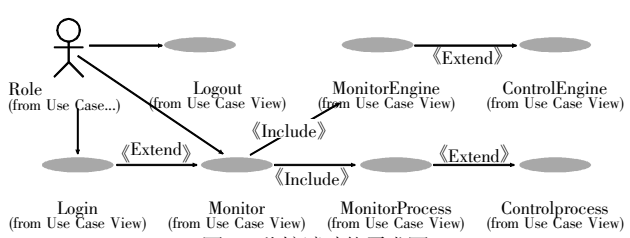


图 4　监控端功能需求图

监控管理客户端分为引擎监控和流程监控,采用 MVC 模式设计实现了系统良好的扩展性,Model 代表应用程序的数据和用于控制访问和修改这些数据的业务规则。当模型发生改变时,它会通知视图( View) ,并且为视图提供查询模型相关状态的能力。同时,它也为控制器( Controller) 提供访问封装在模型内部的应用程序功能的能力。系统中它负责与引擎进行交互,调用引擎所提供的方法,获取相应的数据; View 用来组织模型的内容。它从模型那里获得数据并指定这些数据如何表现。当模型变化时,视图负责维持数据表现的一致性。视图同时将用户要求告知控制器( Controller) ; Controller 负责对来自视图的用户要求进行解释,并把这些要求映射成相应的行为,这些行为由模型负责实现。

监控管理客户端分别对工作流引擎和软件过程流程的状态进行监控与管理,根据状态可以启动、暂停、恢复和停止引擎。同时还可以对流程进行控制,可以实现流程的启动、挂起、恢复和停止等操作,并且还可以实现流程跳转操作,这些操作的实现都是通过远程调用引擎的方法来实现的。其中监控客户端的具体流程如图 3 所示。

### 2.2.3　工具集成客户端

客户端的主要功能是集成了常用的软件开发工具,比如文档工具、设计工具和编码工具。用户接收到任务后可以直接在调用集成的工具进行相关的开发工作。工具的集成的设计思路是: 考虑到系统需求的原因,工具集成采用松耦合的形式,即是通过读取工具的配置文件来获取本地已安装工具的调用信息。工具之间的信息共享采用信息共享中心库的形式,中心库是存储和组织所有与特定软件系统有关信息资源的一种机构。其根本目的是作为存储与特定系统有关的一切信息的场所,保证存入的信息的一致性、完整性和方便的共享性。也就是说中心库除了具有一般 DBMS 的功能外,还要为集成化的 CASE 环境提供下面一些功能支持: 信息一致性、完整性控制; 系统信息实施管理; 文档标准化; CASE 工具集成; 数据与数据集成等。也就是要为数据集成、界面集成、控制集成与过程集成提供一元化的环境支持。系统的中心库是一个文件系统,通过它来实现工具之间的信息共享。

具体实现如下: 用户可以根据自己的需要在本地操作系统中安装合适的开发工具,然后通过客户端的配置界面把这些工具配置到 SPDET 中,系统会自动根据配置生成一个 XML 格式的工具配置文档 toolconfig. xml( 图 5) ,然后根据配置文档生成相应的图形菜单供用户调用工具,用户可以通过工具栏菜单来启动工具,也可以从右键菜单打开。最初的版本中考虑集成进来的工具分成三类,分别是文档工具、设计建模工具和编码工具,也可以根据需要集成软件开发中的其他的相关工具如说配置管理工具( CVS) 、测试工具等。中心库采用文件系统来实现,考虑到扩展性可以在以后扩展到数据库来实现信息共享。

### 2.2.4　建模客户端

工作流建模是用建模语言来描述复杂的软件开发过程。由于软件过程包含了大量的人与人之间、人与机器之间的交互活动以及人的创造性活动,使得软件过程具有高度的复杂性和动态性,因此过程建模活动是最复杂的活动之一。因此采用合适的建模语言来进行软件过程流程到工作流流程的转换是一个关键的问题,介绍的是与用户进行交互的可视化的建模客户端。它采用经典的 MVC 体系结构,建模人员可通过 GUI 图形

用户界面直观、高效地创建过程模型,当操作失误时,工具能给出警告和错误信息。图 6 是建模客户端的功能需求图。

```
<?xml version="1.0" encoding="GB2312">
<tools>
  <tool>
   <tool_name>doc</tool_name>
   <tool_path>
        D:\Program Files\Microsoft Office\OFFICE11\WINWORD.EXE
    </tool_path>
</tool>
<tool>
 <tool_name>code</tool_name>
  <tool_path>
        C:\WINDOWS\NOTEPAD.EXE
   </tool_path>
</tool>
<tool>
  <tool_name>modul</tool_name>
  <tool_path>
        D:/Program Files\Rational\Rose\rese.exe
  </tool_path>
</tool>
</tools>
```

图 5 配置文档 Toolconfig.xml

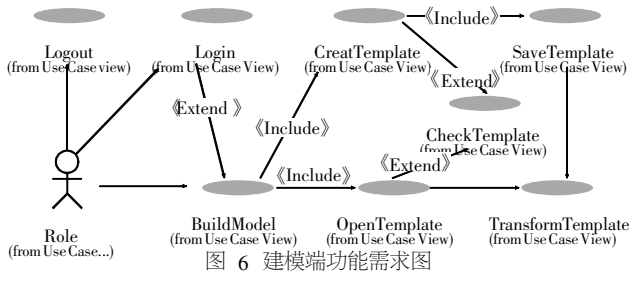


图 6 建模端功能需求图

工具的主要功能是完成过程定义,提供对过程定义的静态语法检查,并将正确的过程定义的数据以 XML 的形式保存在服务器端模板库中。基于这些功能考虑,可将系统划分为三个主模块:过程建模、模型文件格式转换、模型校验。

(1) 过程建模:建模人员可以使用建模图元在编辑区进行建模。

(2) 模型文件格式转换:工作流建模工具使用 XML 格式存储过程定义文件。当存储过程模型时,需要将过程模型的数据结构转换为 XML 格式;当打开过程模型文件时,要将 XML 格式的过程模型文件解析到内存的数据结构中。具体实现过程为:由于工作流建模机制中的图形元素的信息具有一定的结构,与 XML 语法结构相对应,所以采用 DOM, SAX, JDOM 来实现元素信息到 XML 的转换;首先,把图形元素的所有属性表示成 XML 文档的属性文件,然后当一个图形元素的 Editor 类监听到相应的属性发生改变并对它作出响应时,图形元素的属性值将会获得新的值,XMLElement 类就会从属性文件中读出相应图形节点元素的属性,再把获得的新值赋予属性文件中图形节点的属性,再通过 DOM 或 JDOM 解析把元素属性转换成 XML 文档。

(3) 模型校验:模型校验是依据定制的工作流建模语言的语言找出错误,帮助用户及时进行修改,从而定义规范的过程模型。

### 2.3 中间层

工作流引擎(Workflow Engine)是系统中核心的部分,是过程运行的推动机制。工作流引擎模块包括两个部分:构造期引擎(Build-time Engine)和运行期引擎(Run-time Engine)。其中构造期引擎主要解析建模工具生成的 XML 文件,验证工作流模板的正确性和完整性,并把工作流模板及使用到的相关数据导入数据库中。运行期引擎主要负责根据构造期引擎生成的模板与包的数据包和工作流模板相关数据的实例化,按照建模时定义好的流程实际执行模板中的每个节点。

引擎中包括了工作流执行相关的一些功能组件,其中消息管理器在中间层和客户层之间起着纽带的作用,其主要功能是负责整个系统的消息的接收和转发。系统的消息服务架构在 J2EE 的 JMS(Java Message Server)服务之上,开发平台通过 JMS 来实现消息的管理。消息管理器采用主题(Topic)服务的订阅/发布(Pub/Sub)模型给订阅的客户层发送消息。客户层有一个对应的 JMS 客户端异步的接收服务端发送过来的 JMS 消息。同时客户层也需要一个消息服务器来向中间层的消息管理器发送任务完成的消息,但是它采用点对点模型(Point-To-Point, PTP)异步的发送 JMS 消息,在消息管理器中有一个相应的消息队列(Queue)接收任务提交消息,然后消息管理器将消息转换成相应的数据格式提交给引擎或者直接存入数据库中供引擎调用。

### 2.4 数据层

数据层由一个文件系统和一个关系型数据库(DBMS)组成,主要为系统提供持久的存储机制。其中文件系统是用来为系统提供存放过程建模文档和软件开发过程中产生的软件过程文档以及一些规范的开发模板文档等。关系数据库用来存储引擎启动后通过解析模板得到的建模数据、运行时数据和需要保存的一些工作项数据。

## 3 结束语

描述的系统针对软件过程的特点结合工作流的优势,使得软件过程半自动地执行,提高了软件过程的开发效率和质量,同时也减少了软件过程管理的工作量,提高了软件开发的成功率。同时系统的另外一大特色是把软件过程的开发工具整合进来了,从而为软件开发提供了完整的、一致性的过程服务。但是还有一些需要进一步完善的地方:在最初的版本中并没有考虑组织建模,因此对于任务的接受暂时还只能针对角色(假定角色只有一个人)。在后期的开发中会在业务客户端对任务的再次分派做出合理的实现;在工具集成部分需要进一步改善的是集成进来更多的工具,或者考虑把一些小的开源的小工具集成进来。所以系统还有待在实践和理论上进一步提高。

## 参考文献:

[1] Madhavji, *et al*. Guest Editors 'Introduce. IEEE Trans. [J]. On Software Engineering, 1993, 19(12): 1125-1127.
[2] Osterweil, L J. Software Process Are Software Too, Revisited[C]. Proceedings of the 19th International Conference on Software Engineering (ICSE 1997), Boston, MA, 1997.
[3] Daniel K C Chan, Karl R P H Leung. Software Development as a Workflow Process[C]. The 4th Asia-Pacific Software Engineering and International Computer Science Conference(APSEC '97/ICSC '97), 1997. 282-292.
[4] Dowson M. In Iteration in the Software Process[C]. IEEE Comp. Soc. Press, 1987. 29-32.
[5] David Sharon, Rodney Bell . Tools that Bind: Creating Integrated Environments[J]. IEEE Software, 1995, 12(2): 76-85.
[6] 宗志东,等. 软件过程技术研究[J]. 计算机科学, 1997, 24(3): 6-10.
[7] 郭江,黄涛. 软件过程集成环境研究[J]. 计算机科学, 1997, 24(4): 6-13.
[8] 周俊鹏,邹晓宇. 基于软件过程的工作流引擎技术报告[R]. 长沙:国防科学技术大学, 2004.

## 作者简介:

邹晓宇(1979-),男,湖南宁远人,硕士研究生,主要研究方向为计算机软件工程、工作流和软件基础件;王戟(1969-),男,湖南常德人,教授,主要研究方向为计算机软件工程和形式化方法等;周俊鹏(1980-),男,湖南邵阳人,硕士研究生,主要研究方向为计算机软件与理论。