

一种柔性软件框架的设计与实现

赵艳妮^{1,2}, 郭华磊³, 尚 岚², 范秀峰²

(1. 陕西职业技术学院 计算机科学系, 陕西 西安 710100;

2. 西安理工大学 计算机科学与工程学院, 陕西 西安 710048;

3. 西安通信学院 信息服务系, 陕西 西安 710106)

摘 要:针对用户业务流程发生变化,服务组合的灵活性将受到挑战的问题,从柔性软件理论和数据处理流程两个方面进行深入分析,在现有技术的基础上提出了面向服务、基于工作流管理的软件框架。它将业务流程层和服务层抽象出来,利用工作流管理的灵活性实现了一种具有柔性的软件框架技术,真正实现软件研发工业化的构件组装,同时支持数据业务流程的灵活重组。最后,以柔性软件理论为指导,结合某大型制造企业的工时管理系统的研发,详细描述了该框架的具体实现方法和过程。实际应用表明,该框架实现了模块的复用,降低了软件的研发成本,保障了软件产品的质量。

关键词:软件框架;面向服务;工作流;柔性;Petri 网

中图分类号:TP391.5

文献标识码:A

文章编号:1673-629X(2015)11-0093-06

doi:10.3969/j.issn.1673-629X.2015.11.019

Design and Implementation of a Flexible Software Framework

ZHAO Yan-ni^{1,2}, GUO Hua-lei³, SHANG Lan², FAN Xiu-feng²

(1. Department of Computer Science, Shannxi Vocational & Technical College,

Xi'an 710100, China;

2. College of Computer Science and Engineering, Xi'an University of Technology,

Xi'an 710048, China;

3. Department of Information Service, Xi'an Communication Institute, Xi'an 710106, China)

Abstract: According to the problem that the user business process changed, the flexibility of service composition will be challenged, conducting in-depth analysis from two aspects of flexible software theory and data processing, present a software framework for service-oriented and workflow-based on the basis of existing technology. The framework abstracts the process-layer and service-layer, and uses the flexibility of process-management to implement a flexible software framework, realizing the industrialization of software development component assembly, while supporting flexible reconstructing of business process data. Finally, guided by the theory of flexible software, take the design of Time Quota Management of a large manufacturing enterprise as an example to elaborate the realization of this framework. The actual application shows that the framework proposed in this paper implements the module reuse, reducing software development costs, guaranteeing the quality of software product.

Key words: software framework; service-oriented; workflow; flexibility; Petri nets

0 引言

软件的变化性即软件的“柔性”受到了人们普遍关注^[1]。虽然软件体系结构或软件框架是软件“共性”即“刚性”的重要体现,但软件如何适应各种变化是人们目前遇到的难题。怎样实现变化性,从方法上归纳起来主要有两类,一类是通过外围扩展,即变化点的方式来实现变化性^[2-4],另外一类是通过软件自身

结构的变化来实现变化性^[5-6]。总体来看,以上两类方法的研究主要集中在软件体系结构上。与软件体系结构相比,软件框架的易理解性和易实现性,使之更受软件开发人员的青睐,因此文中从软件框架的角度来研究软件的“柔性”问题,并给出了一个柔性软件框架。该框架的核心思想是:基于 SOA(Service Oriented Architecture),将业务流程层抽象出来,结合服务的理

收稿日期:2015-02-05

修回日期:2015-05-06

网络出版时间:2015-11-04

基金项目:国家自然科学基金资助项目(61272284);陕西省教育自然科学基金(2013JK1202);陕西省自然科学基金(2014JM8354)

作者简介:赵艳妮(1982-),女,博士研究生,讲师,研究方向为软件工程、构件测试等。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20151104.0948.020.html>

念,利用 workflow 组合“服务”来实现软件框架的“柔性”,基于此框架开发的软件对变化具有更好的自适应性。

文中首先给出构建面向服务、基于 workflow 管理的软件框架结构,结合业务流程建模和验证方法,通过具体实现技术分析了框架的搭建机理。然后结合某大型制造企业的工时管理系统的开发,详细说明了该框架的具体实现过程。

1 面向服务、基于 workflow 管理的软件框架

软件发生变化的原因有很多:网络环境变化可能影响软件的部署,用户需求的变化可能引起软件业务流程的重新组合。面向服务的体系结构(SOA)将所有功能都定义成相互独立的可复用的服务,根据网络环境调整部署方案,访问分布在集群服务器上的相应服务。面向服务的设计可以解决软件功能性服务的调整,如果业务流程发生变化,服务组合灵活性将受到挑战。将业务流程层抽象,结合 workflow 管理模式^[7],使功能性服务以自适应的方式组合,灵活应对业务流程的变化。

文中以 workflow 和服务为基础,给出一种基于 workflow 管理、面向服务的软件框架,如图 1 所示。

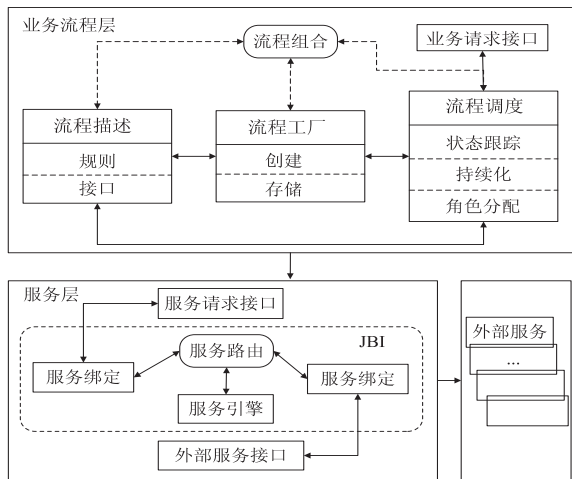


图 1 面向服务、基于流程管理的软件框架

该框架的柔性主要体现在应对各种变化而给出的一种软件框架层面的自适应性,结构上由业务流程层和服务层组成。其中,业务流程层通过业务接口向用户提供统一的业务调用服务,并通过流程调度模块和流程描述模块对服务请求接口进行封装。服务层对现

有服务进行封装和管理,向上层业务流程层提供统一访问接口。

该框架的基本运行机理^[8]是:业务接口模块接收用户请求,流程工厂模块创建流程实例注入到流程组合模块,流程组合模块对流程进行组合。流程调度模块根据用户请求对请求流程进行状态跟踪,激活流程组合模块提供的流程实例。流程描述模块对此流程进行标准化描述,为下层服务层提供统一访问接口。服务层接收来自业务流程层的服务请求,服务绑定模块对外部服务接口接收或者发送消息。通过服务路由模块向服务层内部提供统一的消息传递功能。服务引擎模块不与外部通信,它负责接收到消息后,对消息做一些处理(比如协议的转换,消息格式的变化,各种服务的集成等),然后由服务绑定模块调用外部服务。

软件的“柔性”体现在其适应变化的能力,包括功能变化、数据变化,甚至用户新的需求引起流程的变化等。该软件框架虽然借鉴了一些现有技术,但这些技术尚不成熟,仍处在不断发展的过程,无法完全满足软件柔性的要求,无法从根本上解决适应变化的问题。在柔性软件的开发中需要对相关技术进行整合,协调互动。因此,需要从框架的内部构造方面对柔性进行分析,确定软件柔性设计的体现,从而探索框架各部分的设计方案,为最大限度地适应软件变化的可行性提供依据。

1.1 业务流程层

在一个应用系统中,业务流程是比较容易变化的,基于传统的三层框架设计不涉及业务流程的管理,一旦业务流程发生了变化,则每层的改动都非常大。基于流程设计的项目与传统项目的一个最大区别就是前者可以随时改动业务流程,而整个系统不需要太大的改动甚至不需要改动代码就可快速恢复使用。其原因在于,面向服务、基于 workflow 管理设计的业务流程层可以快速组合改动的业务流程,服务则是以“可插拔”的形式提供应用系统调用。这即是该框架“柔性”特征的主要体现。所以,将业务流程管理和服务两者进行分离是面向服务、基于 workflow 管理的软件设计框架可以顺利实施的关键。这样一来,两者之间就处于一种松散耦合的状态。

图 2 说明了业务流程层中流程实例的创建和各个模块对流程管理的整体过程。

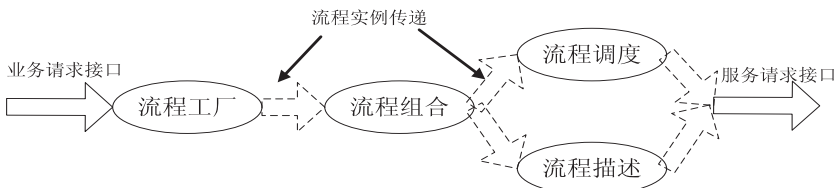


图 2 业务流程层流程管理的整体过程

其中,流程组合模块是使业务流程层具有柔性的关键。它接收流程工厂注入的流程实例进行一系列的组合操作,使流程实例能够满足系统需求。同时流程调度模块和流程描述模块对流程实例进行辅助操作使系统得知流程的状态和表示。

1.1.1 流程组合模块

针对业务流程的易变性,解决方法是使业务流程具有一定的灵活性。如图 3(a)所示,一个圆圈表示一个独立的服务,若在系统中有相对独立的几条业务流程,由其中的服务组合而成,1-2-3-4、6-3-4、5-2-3-4 都是可独立的业务流程,业务流程原本有多个入口(1、5、6)和出口 4,但中间又有重叠,可以将它们整合成如图 3(b)的业务结构。使系统的入口和出口统一,需要调用哪个服务由输入参数而定,不需要调用的服务可以将它轮空,这相当于将原来固定不变的业务流程分解成若干个小片段,再由流程组合模块这条“线”给灵活链接起来。这样便可以使整个业务流程具有通用性,对于后期维护和推广非常有利。

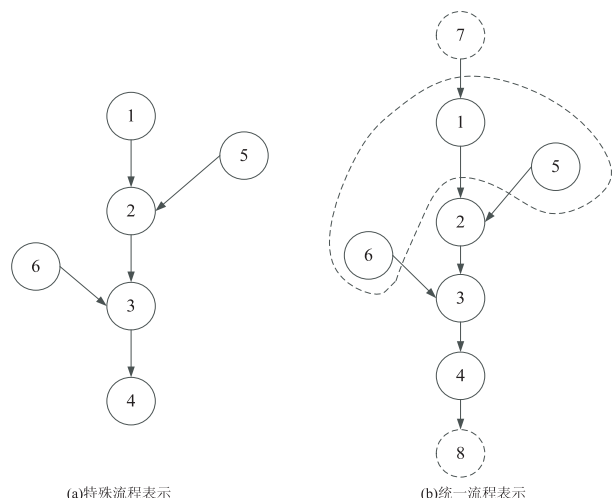


图 3 流程的灵活组合

1.1.2 流程工厂模块

流程工厂负责流程的查询、创建和持久存储。根据业务请求接口创建流程实例。流程组合模块在实例化流程对象时,可通过注入控制反转来动态调用流程工厂模块中的流程实例。流程调度模块可根据该模块查找找到可以激活的流程或是重新开启新的流程,这可由配置文件和抽象工厂来完成。组合模块从配置文件中读取流程名称,由抽象工厂实例化流程实例并注入到组合模块中。这样的话,如果需要动态组合流程,便可实现两模块之间的解耦。

1.1.3 流程调度模块

流程调度模块根据用户请求对请求流程进行状态的跟踪,它为系统提供流程实例运行的历史记录。如果流程的一次完整执行允许中间某些步骤之间有时间中断,状态跟踪功能就是记录一个流程实例每次的具

体运行情况。持续化功能支持流程的异步操作。如果流程需要几个角色的协作来完成,也需要持续化功能记录流程实例的具体状态。角色分配功能根据用户角色为流程组合模块指定某个角色的功能组限制。以上三个功能都可放在数据库中实现。比如持续化功能将未执行完成的流程实例中的各个活动状态以文件或数据库表方式存储,待需要时再根据状态跟踪功能恢复运行此流程实例的线程。

1.1.4 流程描述模块

这一模块对本层流程调度模块进行流程描述的支持,根据规范的描述可跟踪到具体节点,以便提供流程持续化操作。可用标准规范业务流程执行语言(Business Process Execution Language, BPEL)实现。BPEL 是一种能够将 Web 服务串起来的业务流程语言,它将流程规则描述为 .bpel 文件,对外接口描述为 .wsdl 文件,通过一系列配置文件读取所需信息。对上层用户主要通过与业务接口模块和流程调度模块的协作来对用户操作做出反应。对下层服务层,通过节点描述来定位服务的调用。

BPEL 一般需要在 BPEL 引擎下运行^[9]。一个 BPEL 流程会创建一个 BPEL 流程实例,它将通过合作伙伴连接来实现服务的调用。BPEL 的合作伙伴连接只定义所要调用的接口,这个接口在 BPEL 运行时再绑定到真正的服务提供者上面。总的来说,BPEL 通过流程编程将各种接口组合在一起,目的就是提供一个“集成了各种接口”的表示流程的接口。

1.2 服务层

1.2.1 服务请求接口

服务请求接口是业务流程层与服务层交互的接口,其提供统一的接口供业务流程层调用。服务请求接口描述了服务层可以使用的服务操作,通过构建服务请求接口,业务流程层仅需调用服务请求接口中的接口方法就能获取服务操作,而与接口方法的具体实现无关,实现了业务流程管理与服务管理的分离,使得业务流程的变更不会对服务层产生影响。

业务请求接口是整个系统业务处理的公共入口。通过构建业务请求接口,接口调用者仅需调用接口方法完成业务功能,而不必关心接口方法的具体实现,实现了业务设计与业务功能具体实现的有效分离,使得业务功能实现的调整或修改不会对它的调用者产生影响,增强了系统的可维护性和可扩展性。

1.2.2 服务路由模块

此模块提供了智能路由、协议中介和其他转化机制。它可由基于 JBI 框架的 ESB 工具实现。这种 ESB 运行机理^[8]是将所有的请求信息都送到服务绑定模块 BC,BC 将非规格化的消息转化成规格化信息,再送到

服务路由模块。如果需要对消息进行进一步的处理(比如协议的转换,消息格式的变化,各种服务的集成等),将它送往服务引擎模块,在这个模块里做一些消息格式的处理,再送回服务路由模块,返回服务绑定模块,再转给外部服务。

这里引入 ESB 的主要作用是解除因消息格式或传递协议不同而造成的服务调用问题。因为服务很可能是由外部资源得到的,修改它们的内容比较困难。服务总线使它们能够以“可插拔”的形式提供应用系统调用。

1.2.3 外部服务接口

外部服务接口主要通过 .wsdl 文件来描述,通过一系列配置文件读取所需信息。它提供一种公共的语言表示方法使服务调用者能够找到所需要的服务,定义接口参数等对服务进行抽象的描述。流程可以有不同的组合方式,统一通过流程组合模块的接口调用。组合好的业务流程由流程调度模块调度,将结果返回给用户。

2 应用框架实例

上面给出了面向服务、基于 workflow 管理的软件框架的设计思路。下面结合某大型制造企业的工时管理系统的设计和开发,来阐述这种技术的具体设计过程。

2.1 用 Petri 网对业务建模和验证

Petri 网的概念最早是在 1962 年 Carl Adam Petri 的博士论文中提出。其主要特征包括:并行、不确定性、异步和分布描述能力以及分析能力。

在 Petri 网系统图形表示中,状态元素称为库所(place),用“○”表示;变迁元素(transition)用“□”表示;流关系(arc)一般用“→”表示。

用 Petri 网对业务流程进行建模,因为它可以对业务建模的正确性进行验证,更重要的是,Petri 网作为研究热点理论上可以实现与 BPEL 的转换^[10-12],这对框架中的业务流程层的搭建十分有益。所以对其中一个重要的功能模块—标准工时管理进行 Petri 网建模,同时验证它的正确性。

标准工时管理的业务流程是这样的:分厂定额员制定标准工时,制定完成后上报给财务部的主管定额员,主管定额员对上报的标准工时进行审批,若是不通过审批条件,返回给分厂定额员,若是通过审批条件,则给分厂定额员下发修正系数和审批意见,分厂定额员拿到修正系数后对标准工时进行修正,紧接着进行归档工作。归档之前系统要对标准工时进行检测,如果超出了可归档范围,则需要分厂定额员重新对这一批标准工时进行修正,如果归档工时总数在可归档范围之内,则可以直接归档,流程结束。

图 4(a)所示的是上述业务流程的 Petri 网建模模型。其中,活动库所含义如表 1 所示,变迁含义如表 2 所示。

表 1 库所含义表

库所	含义
P ₁	分厂定额员未制定标准工时
P ₂	分厂定额员制定标准工时
P ₃	主管定额员收到标准工时未审批
P ₄	主管定额员审批标准工时后
P ₅	主管定额员制定标准工时修正系数
P ₆	分厂定额员收到标准工时修正系数
P ₇	标准工时系数被分厂定额员修正后
P ₈	归档之前系统要对标准工时进行检测
P ₉	标准工时归档

表 2 变迁含义表

变迁	含义
T ₁	分厂定额员制定标准工时
T ₂	分厂定额员上报标准工时
T ₃	主管定额员审批标准工时
T ₄	主管定额员不通过的标准工时
T ₅	主管定额员通过的标准工时
T ₆	主管定额员下发标准工时修正系数
T ₇	分厂定额员对标准工时进行修正
T ₈	分厂定额员对标准工时进行归档
T ₉	系统判断归档工时总数超出可归档范围
T ₁₀	系统判断归档工时总数在范围之内,直接归档

模型建立后,首先要验证模型的可行性。目前主流的验证方法是对 Petri 网进行化简后验证。在文献[13]中提出,如果根据这些规则化简过的 Petri 网模型最终是只含有一个库所的话,则这个模型就是可行的。

然后,对模型进行扩展,如图 4(a)所示。依据文献[11]所述的化简规则对该模型进行化简,过程如下:

(1)根据顺序库所化简规则将 P₁、P₂ 合并,根据顺序变迁化简规则将 T₅、T₆ 合并,T₇、T₈ 合并,如图 4(b)所示;

(2)根据顺序库所化简规则将 P₃、P₄ 合并,P₆、P₈ 合并,T₇₋₈ 消去,如图 4(c)所示;

(3)根据变迁自环化简规则将 T₄、T₉ 消去,如图 4(d)所示;

(4)根据顺序库所化简规则将模型化简成为一个顺序库所,如图 4(e)所示;

(5)根据顺序库所化简规则将模型化简为一个库所,如图 4(f)所示。

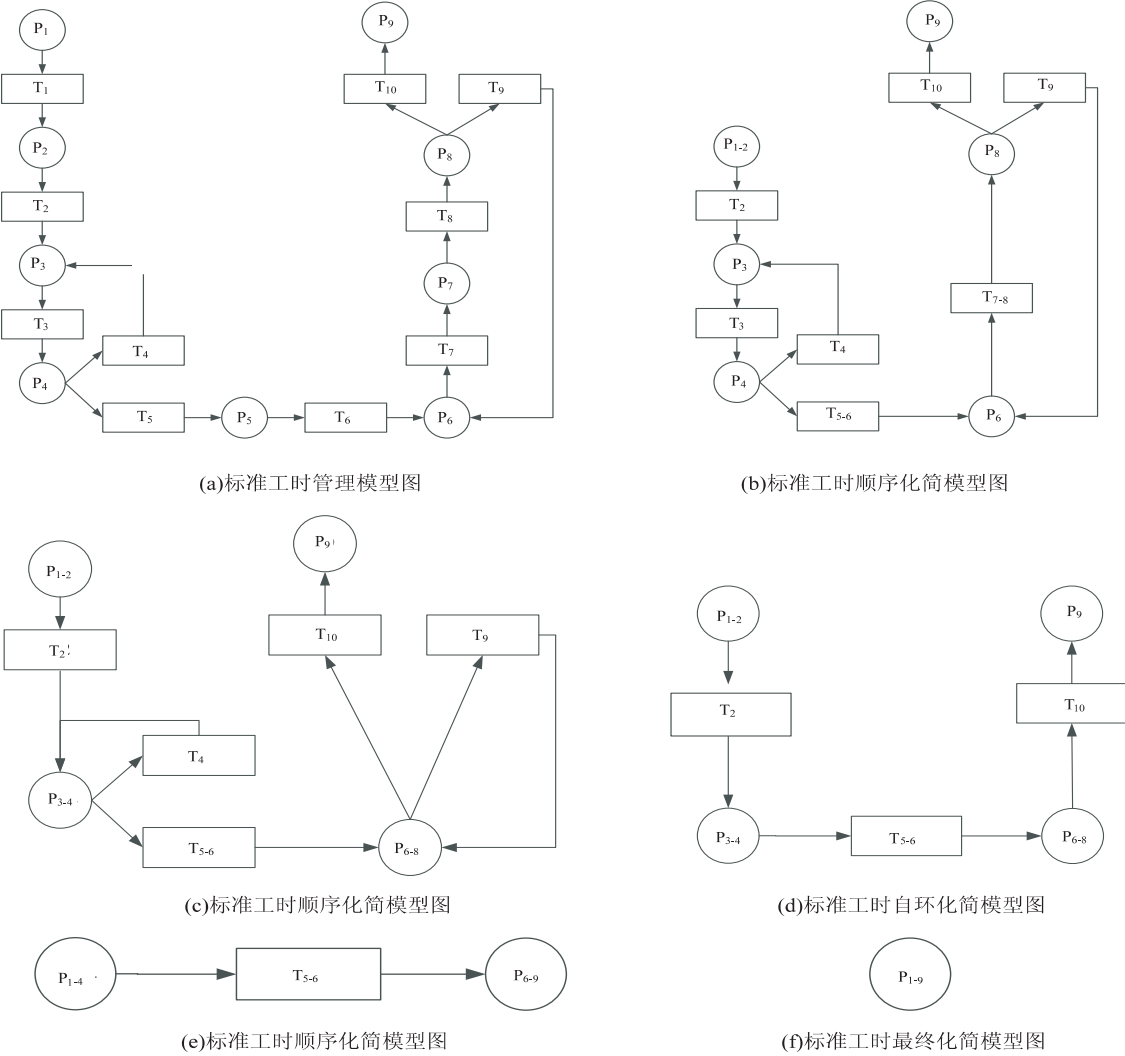


图 4 模型化简结构图

通过以上化简,最终模型只包含一个库所,验证了模型的可行性。

2.2 业务流程层的设计

对业务流程建模后,如果将 Petri 模型的每个变迁看作一个 Web 服务,表示一个相对独立的功能模块,在业务流程层灵活地组合可以得到需要的若干业务流程^[14]。比如现行工时管理、工装工时管理、零星工时

管理等业务流程与标准工时的管理流程有相似,又具有自己的特点。如图 5 所示,针对某个业务流程组合与它相关的功能性服务,对于有变化的服务,业务流程层的自身柔性在某种程度上适应系统的变化,几乎不需要改动代码。该软件框架的复用性和可维护性都得到大幅度的提高。

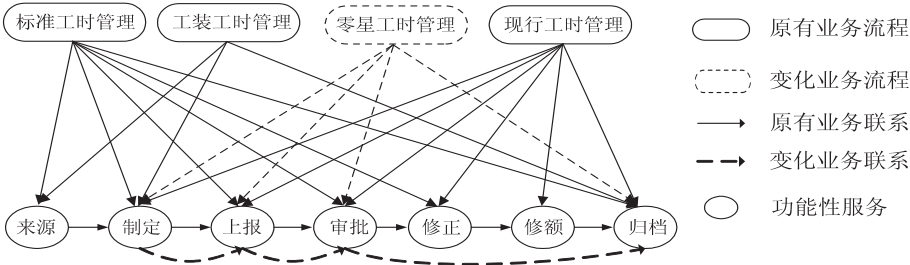


图 5 实例业务表示

工时管理流程包括两种可能的流程执行步骤,要求它们在柔性的软件框架里都可以得到动态实现。针对同一业务的处理可以有多种操作方法,将它们分别以实线和虚线表示。同为对工时的管理这一条业务,

不同的工时或者不同专业厂对它的处理并不完全相同。这种情况在大型企业中是常见的。那么在柔性的软件框架下开发的软件特点即是将这些不同点考虑在内,当遇到这些变化时,可以提供自适应的调整。解决

思路是使流程具有一定的可变动性,即是流程的各部分信息可在运行时动态获得,这样便可以通过对相同接口的封装来满足不同流程的运行。

以流程实体为例,一条流程是由很多步骤组成的。这步骤的实现是通过调用 Web 服务来实现的。这个步骤的数量因为软件柔性的要求是可变的,可由用户和开发者动态调控。而每一个步骤需要定义所调用的 Web 服务,输入的参数以及返回的对象。这些都是编程期间所不能确定的,可利用反射来实现。

在实际的业务流程中,流程需要组合,区别是组合的具体方法不同。采用 Template 模式可以实现业务流程的灵活组合管理,因为 Template 模式仅仅定义组合算法的骨架,而将具体步骤延迟到子类中,子类可以不改变组合算法的结构即覆盖该算法的某些特定步骤,即子类可以根据具体情况采用不同的方式来实现抽象类或接口中抽象的通用操作,灵活地组合出所需的业务流程来。

2.3 服务层的设计

服务层的主要作用是在外部服务之间进行调配,使它们以一种统一的方式可供流程管理层调用。在具体实现工具上,基于 JBI 的 ServiceMix 是一个轻量级的开源 ESB,通过它可以实现软件框架中服务层的搭建。

ServiceMix 利用一系列配置文件来实现调配。其中,SA 目录下的配置文件 jbi.xml 负责配置运行在 ServiceMix 运行环境中的服务集成模块;SU 是服务单元对复杂应用进行单元划分,每个 SU 将会打包成.zip 包集成到 SA 包中运行。SU 目录下的 servicemix.xml 文件配置服务单元,根目录下的 servicemix.xml 运行服务。由于此技术相对比较成熟,通过测试可以在软件框架中正常运行。

3 结束语

软件框架近年来受到普遍的关注和应用,但软件框架的“柔性”问题是软件框架应对多样变化特别是网络环境变化中遇到的主要问题。文中基于 SOA 和工作流理论给出的框架较好地解决了这一问题,并结合某大型制造企业的工时管理系统的开发,使该框架得到充分的验证,达到了预期的效果。文中给出的柔

性软件框架及其实现方法,对一般软件的开发具有普遍的适应性。

参考文献:

- [1] 王映辉. 软件功能需求变化传播机理分析[J]. 计算机学报,2007,30(11):2025-2032.
- [2] 段 华. 基于扩展 BPEL 模型的 SOA 流程生成[D]. 北京:北京邮电大学,2012.
- [3] 申利民,齐 硕. 面向用户柔点的设计模型[J]. 计算机工程与设计,2011,32(6):2024-2126.
- [4] Rettkowski J, Wehner P, Schülper M, et al. A flexible software framework for dynamic task allocation on MPSoCs evaluated in an automotive context[J]. LNCS,2015,9040:255-266.
- [5] Ciobanu G, Juravle C. Flexible software architecture and language for mobile agents[J]. Concurrency and Computation: Practice and Experience,2012,24(6):559-571.
- [6] 丁 博,王怀民,史殿习,等. 一种支持软件可信演化的构件模型[J]. 软件学报,2011,22(1):17-27.
- [7] 杨志伟,吴 兵. 基于 UML 状态图的软件测试充分性准则研究[J]. 计算机技术与发展,2013,23(8):43-46.
- [8] Mennicke S, Lochau M, Schroeter J, et al. Automated verification of feature model configuration processes based on workflow Petri nets[C]//Proceedings of the 18th international software product line conference. [s. l.]:ACM,2014:62-71.
- [9] Baresi L, Guinea S. Self-supervisingBPEL processes[J]. IEEE Transactions on Software Engineering,2011,37(2):247-263.
- [10] 顾 军,罗军舟,曹玖新,等. 基于排队 Petri 网的服务系统性能建模与分析方法[J]. 计算机学报,2011,34(12):2435-2455.
- [11] Pla A, Gay P, Meléndez J, et al. Petri net-based process monitoring: a workflow management system for process modelling and monitoring[J]. Journal of Intelligent Manufacturing, 2012;25(3):1-16.
- [12] 吴育春,李蜀瑜. 基于时间 Petri 网的 AADL 模型[J]. 计算机技术与发展,2014,24(2):88-91.
- [13] Hsieh F S, Lin J B. Development of context-aware workflow systems based on Petri Net Markup Language[J]. Computer Standards & Interfaces,2014,36(3):672-685.
- [14] Heilig L, Voss S. A cloud-based SOA for enhancing information exchange and decision support in IIT operations[M]//Computational logistics. [s. l.]:[s. n.],2014:112-131.

一种柔性软件框架的设计与实现

作者：[赵艳妮](#)，[郭华磊](#)，[尚岚](#)，[范秀峰](#)，[ZHAO Yan-ni](#)，[GUO Hua-lei](#)，[SHANG Lan](#)，[FAN Xiu-feng](#)

作者单位：[赵艳妮, ZHAO Yan-ni \(陕西职业技术学院 计算机科学系, 陕西 西安 710100; 西安理工大学 计算机科学与工程学院, 陕西 西安 710048\)](#)，[郭华磊, GUO Hua-lei \(西安通信学院 信息服务系, 陕西 西安, 710106\)](#)，[尚岚, 范秀峰, SHANG Lan, FAN Xiu-feng \(西安理工大学 计算机科学与工程学院, 陕西 西安, 710048\)](#)

刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：[2015, 25\(11\)](#)

引用本文格式：[赵艳妮](#). [郭华磊](#). [尚岚](#). [范秀峰](#). [ZHAO Yan-ni](#). [GUO Hua-lei](#). [SHANG Lan](#). [FAN Xiu-feng](#) [一种柔性软件框架的设计与实现](#)[期刊论文]-[计算机技术与发展](#) 2015(11)