

# 一种生产线架构中的工作流过程定义复用方法

龚晓庆 刘 锋 葛 玮 郝克刚

(西北大学信息科学与技术学院 西安 710069)

**摘 要** 针对 workflow 管理系统在实际应用中面临的过程定义低效、复杂的问题,借鉴软件复用的思想,提出了一种生产线架构下的工作流过程定义复用方法。该方法建立在领域业务本体和工作流模板这两类可复用资产之上,尝试建立特定领域工作流过程定义的复用机制,以提高过程定义的效率。着重探讨了可复用资产的构建、描述和检索等问题,介绍了基于复用的工作流过程定义方法,并开发了相应的实用工具以验证该方法的有效性。

**关键词** 软件复用,软件生产线,工作流,领域业务本体,工作流模板

**中图法分类号** TP311.5 **文献标识码** A

## Workflow Process Definition Reuse Method in Software Product Line Architecture

GONG Xiao-qing LIU Feng GE Wei HAO Ke-gang

(School of Information Science and Technology, Northwest University, Xi'an 710069, China)

**Abstract** There is a problem in practical application of workflow management system. It is that the workflow processes have been defined with a low efficiency and productivity due to the complexity of actual business processes. An approach to this problem was proposed by adopting the idea and technique of software reuse. The solution is based on the architecture of software product line and aims to improve the efficiency of the domain-specific workflow processes' definition. Two types of reusable assets, domain business ontology and workflow template, were identified. Moreover, how to construct, describe and retrieve these reusable assets was discussed in detail. The steps to define the workflow process based on reuse were explained. As a demonstration, a definition tool was developed and the method was proved feasible and more productive in practice.

**Keywords** Software reuse, Software product line, Workflow, Domain business ontology, Workflow template

工作流技术是实现企业业务过程重组、过程管理与过程自动化的核心技术,工作流过程定义是该领域的一个重要研究课题。工作流过程定义就是将业务过程描述为可以由计算机处理的形式化表示,其中明确描述了业务过程中的活动及其相互关系,还定义了活动之间传递的信息、执行活动的应用程序、活动所需的资源等。要完整地描述一个业务过程,需要对其从不同角度详尽了解,因而在实际应用中,过程定义是一项耗时而且困难的工作。

本文从软件复用的角度对工作流过程定义方法进行了研究,在软件生产线的架构下,提出了一种面向领域的过程定义复用方法,探讨了两类可复用资产——领域业务本体和工作流模板的构建、描述和检索等问题。采用该方法,不必从零开始定义业务过程,而是复用过去积累的领域知识和经验,从而可以降低过程定义的复杂度,提高过程定义的效率和质量。

## 1 工作流技术与软件复用

### 1.1 工作流技术应用中的过程定义问题

工作流管理系统(也称为过程管理系统)被称为是继数据库管理系统和人机界面管理系统之后,第三次从应用中分离

出来的应用系统不可或缺的系统。在工作流技术领域,主要研究内容包括:工作流管理系统体系结构、工作流过程定义与工作流过程定义语言、工作流的实现技术、工作流的互操作技术、工作流的仿真与分析等<sup>[1]</sup>。基于工作流管理系统开发应用程序时,首先要由过程定义工具完整地描述业务过程,而后工作流运行服务就可以自动或半自动地执行该过程。能否准确、快速地进行过程定义关系到是否能够满足企业的业务需求,其直接影响业务过程的运行效率。

西北大学软件工程研究所多年来一直进行工作流过程定义方面的研究,提出了信牌驱动式工作流计算模型<sup>[2,3]</sup>。基于该过程模型,研究所与西安协同数码公司合作开发了基于 Web 的工作流管理系统 SynchroFlow。作为自主知识产权的工作流中间件平台, SynchroFlow 已经在办公自动化、电子政务、电子商务等领域成功应用。

在实际应用中,发现了几个问题:(1)应用中需要的业务过程数量相当大,以某市的电子政务系统为例,仅“审批”业务过程就有 600 多个。(2)过程定义是一项耗时而且困难的工作,完整地描述业务过程涉及到资源、信息等多个方面的视图,需要付出大量劳动。(3)过程定义是一项依赖于经验的工

收稿日期:2008-12-31 返修日期:2009-09-07 本文受 863 国家重点基金项目(2007AA010305)资助。

龚晓庆(1974—),女,博士,副教授,主要研究方向为软件工程等,E-mail:gxq@nwu.edu.cn;刘 锋(1979—),男,讲师,主要研究方向为软件工程等;葛 玮(1960—),男,副教授,主要研究方向为中间件技术等;郝克刚(1936—),男,教授,主要研究方向为软件工程与软件理论。

作,有经验的开发者进行过程定义时效率明显高于新手,原因是他们曾经进行过类似系统的过程定义,比较了解领域业务知识,还能够凭记忆在以前的项目中寻找可复用的资产,如已有的过程定义、表单和 jsp 页面等。

针对上述问题,考虑如果能将软件复用的思想引入到工作流过程定义,为工作流过程定义提供系统化的复用机制,将会大大提高过程定义的效率。

## 1.2 软件复用和软件生产线

软件复用是指重复使用为了复用目的而设计的软件的过程。实践证明,软件复用能够从根本上提高软件生产率,降低开发成本,增强软件质量。目前主流的软件复用技术有面向对象的软件复用、基于构件的软件复用、基于 COTS 的复用和面向生产线的复用。

软件复用的经验显示:领域具有内聚性和稳定性,可复用信息具有领域特定性,面向领域的软件复用容易取得成功<sup>[4]</sup>。领域是指一组具有相似或相近软件需求的应用系统所覆盖的功能区域。基于构件的软件复用技术提供了通用的构件库,通过检索、选取构件库中的构件并组装到新系统架构中来构造软件。但是,在开发相同领域的类似产品时,构件库却不足以表达领域的复用特征,而软件生产线解决了这个问题。生产线工程是一种特殊形式的软件复用,用于开发相同领域的类似产品。它指导开发组织利用生产线的核心资产开发产品,而不是一个接一个地从零开始开发产品。生产线工程强调针对特定领域产品家族的领域分析和面向复用的开发。

本文的研究目的是提供一种特定领域工作流过程定义的复用机制,生产线工程的思想为本文的研究提供了重要的指导。本文的工作是在国家 863 重点课题“面向流程管理的软件生产线”的框架下进行的。该课题提出了面向流程管理的软件生产线架构,如图 1 所示。整个生产线由软件资源库、生产线集成运行环境和生产线工具集、生产线资源可信评估工具集和生产线可信评估工具集几部分构成。本文着重探讨了以下几个问题:(1)特定领域可复用资产的生产、描述和检索;(2)基于可复用资产的工作流过程定义方法;(3)基于复用的过程定义工具。

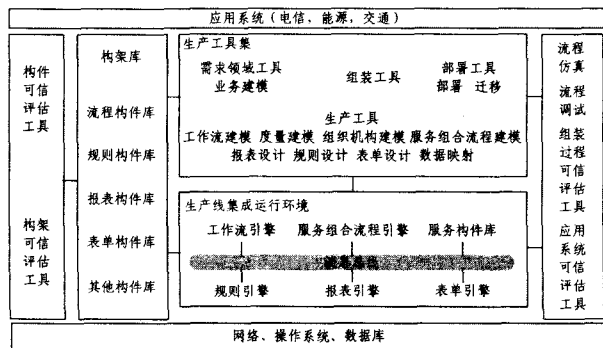


图 1 面向流程管理的软件生产线架构

## 2 特定领域的可复用资产

工作流过程定义涉及到分析、设计与优化业务过程,是一项依赖于经验的工作。文献[5]提出的工作流模板提供了通用业务过程的标准化定义,设计者可以根据实际需求对其进行剪裁和修改从而产生新的过程定义。但是,利用工作流模板进行过程定义仍然需要理解领域业务,否则可能导致选择

不适当的模板。如果领域中通用的业务知识能形式化地表达用以共享,那么开发人员理解领域的效率与质量可以得到提高,同时也为工作流模板提供了应用语境,引导工作流模板的复用。

基于以上考虑,本文从两个层面上考虑了过程定义的复用。在实现层次上,利用工作流模板作为通用业务过程的标准化定义,并探讨了工作流模板的表示、检索等问题。在概念层次上,利用领域业务本体帮助开发人员理解领域业务,捕获具体系统的业务需求,促进特定领域通用业务知识的共享,并且为工作流模板提供语义描述。领域业务本体和工作流模板形成了过程定义中两个不同层次的可复用资产,领域业务本体表示了领域的问题空间知识,而工作流模板是对某个业务功能的具体实现。基于软件生产线的工作流过程复用方法就建立在这两类可复用资产的基础上。下面说明这两类可复用资产的构建、描述和检索方法。

### 2.1 领域业务本体

“本体是共享概念模型的明确的形式化规范说明”<sup>[6]</sup>。构建本体的目的是为了实现在某种程度的知识共享和复用。我们提出领域业务本体的目的是帮助开发者从宏观上理解特定领域的业务知识,并用其描述工作流模板的语义,引导开发者获取体现了业务过程具体实现的工作流模板。领域业务本体中并不刻画业务过程具体行为方面的知识,因为领域中同一业务过程可能存在多种设计方案,对应不同的工作流模板。因此,本文主要从以下几个方面刻画特定领域的业务:(1)业务功能,描述领域中需要完成的工作或任务是什么。业务功能可自顶向下分解为层次结构,由业务过程或活动实现。(2)资源,业务功能执行时被使用或产生、消耗的实体,包括信息、人员、工具等。(3)组织单元,根据组织职能和结构设计的元素,如机构、部门等。(4)角色,对人员按所承担职责的划分。

根据以上分析,可抽象出业务功能、信息、组织单元和角色这几个顶层概念类型。对业务功能进行分解能进一步得到原子业务功能和复合业务功能。识别出概念所属的类型集合之后需要建立概念件的关系,一些概念之间的语义关系定义如表 1 所列。

表 1 一些概念之间的关系

关系	形成规则	说明
part-of	业务功能→业务功能 组织单元→组织单元	整体部分关系,表明业务功能、组织单元可以分解
is-a	业务功能→业务功能 角色→角色 信息→信息……	表示角色、信息等概念间的泛化关系
performer	原子业务功能→角色 原子业务功能→组织单元	表示参与业务功能执行的角色或组织单元
input	信息→原子业务功能	表示为执行业务功能输入的信息
output	原子业务功能→信息	表示执行业务功能后输出的信息

概念类型及概念间关系的定义形成了一个描述业务知识的元模型,规范化、系统化了特定领域中业务知识的描述,领域业务本体是它在特定领域的实例化。

### 2.2 领域业务本体的构建和表示

领域业务本体的构建必须在深入理解特定领域的业务知识的基础上进行。领域业务知识的来源主要有:领域专家、领域相关的业务文档、领域中已成功应用的工作流系统。

构建领域业务本体是领域业务知识概念化和形式化的过

程,其任务包括:

(1)建立概念和概念之间的关系。自顶向下的方法:①识别领域中的顶层业务功能,将其加入概念集;②对可分解的业务功能进行分解,将分解得到的业务功能加入概念集。对其中的原子业务功能分析输入、输出信息、角色、组织单元,将这些概念也加入概念集,并建立它们与业务功能和其他概念的关系;③对可分解的业务功能重复②的操作。自底向上的方法:从各种知识来源中提取有关概念加入到概念集合中,对集合中的概念归类、确定其相互关系,自底向上反复抽取、归并概念及其关系获得领域业务本体。

(2)定义概念的属性。对概念集合中的每个概念定义属性,确定属性取值的类型以及属性约束。这个步骤和上一步骤相互交织、同时进行,是领域业务本体构建中最重要的两步。

(3)使用 OWL 语义表示领域业务本体。采用 OWL 描述领域业务本体,限于篇幅,在此不再举例说明。

### 2.3 workflow 模板

workflow 模板是实现了某业务功能的 workflow 过程定义的封装,外在地等同于一个活动节点。具体地说,workflow 模板实际上是一个参数化的过程定义,其目的是为了复用。workflow 模板的实现中保留了一些与具体系统相关的待定参数,例如活动的具体参与者等,复用 workflow 模板时根据具体情况予以指定,这就是实例化 workflow 模板的过程。本文将 workflow 模板和领域业务本体相结合,实现一个业务功能的 workflow 模板与领域业务本体中某个业务功能类型的概念相关联,workflow 模板的输入、输出数据、参与的角色、组织单元都引用领域业务本体中的概念进行描述。这样就通过领域业务本体中的概念刻画了 workflow 模板的语义。具有语义的 workflow 模板由以下部分组成:

(1)描述:①基本信息描述,包括 workflow 模板的名称、ID、作者、提交日期、版本、描述等;②语义描述,包括一个四元组  $\langle \text{function}, \text{input}, \text{output}, \text{participant} \rangle$ ,其中, function 对应于领域业务本体中某个业务功能类型的概念; input 和 output 是领域业务本体中信息概念类型的概念集合,体现了输入、输出数据的接口的语义; participant 是领域业务本体中组织单元、角色概念类型的概念集合,是 workflow 模板涉及的参与者。

(2)接口:① workflow 模板输入、输出数据的接口;②参数化接口: workflow 模板实现中的待定参数,实例化时指定。

(3)实现:某业务过程形式化描述的文本,其中保留了一些待定参数。实例化时替代参数,形成具体的过程定义。本文使用信牌驱动的工作流模型,业务过程定义转换为 XML 文本保存。由于 workflow 模板的实现外在地等同于一个活动节点,因此在信牌驱动工作流模型中加入一类新的活动节点就可以实现 workflow 模板的嵌套。

本文用 XML 描述 workflow 模板,由于其 XML Schema 的篇幅较长,此处不再给出。

### 2.4 workflow 模板的生产和检索

workflow 模板的生产一般由领域工程师来完成。针对领域中曾经成功应用的工作流项目的过程定义,对其进行抽象,将与具体系统相关的部分定义为参数(如活动的参与者),从而形成 workflow 模板。

特定领域中 workflow 设计者可以根据下面两种方式从库中

检索相应的工作流模板。

(1)关键字检索:根据基本信息的关键字,如作者、入库时间等检索满足条件的工作流模板。

(2)基于领域本体的语义匹配:由于 workflow 模板入库时利用领域业务本体中的概念描述了其语义信息,因此可以通过计算用户查询的关键字在领域业务本体上和 workflow 模板语义描述信息的相似度来检索 workflow 模板。本文采用的语义相似度计算方法如下<sup>[7]</sup>:

设  $t_1$  和  $t_2$  是本体中的两个概念,  $\text{Sim}(t_1, t_2)$  表示这两个概念之间的相似程度,则有公式:

$$\text{Sim}(t_1, t_2) = \sum_{i=1}^n \delta_i(t_1, t_2) \theta_i$$

其中,  $n$  是术语  $t_1$  和  $t_2$  在本体中 kind-of 关系的层次中所具有的最大深度;  $\theta_i$  是权重(可简单地取  $\theta_i = 1/n$ ); 当  $t_1, t_2$  前  $i$  个父类相同时,  $\delta_i(t_1, t_2) = 1$ ; 当  $t_1, t_2$  前  $i$  个父类不相同,  $\delta_i(t_1, t_2) = 0$ 。Sim 返回  $[0, 1]$  间的实数来表示两个概念间的语义相似度。

两个概念集合间的相似度计算方法:设  $M = \{m_1, m_2, \dots, m_x\}$  和  $N = \{n_1, n_2, \dots, n_y\}$  是两个概念的集合,  $\text{SimOfSet}(M, N)$  表示这两个概念集之间的相似程度,则有公式:

$$\text{SimOfSet}(M, N) = \sum_{i=1}^x \sum_{j=1}^y \text{Sim}(m_i, n_j)$$

基于领域业务本体检索 workflow 模板的方法为:用户查询时输入一个概念集合,对于库中的每个 workflow 模板的语义描述的概念集,利用公式 SimOfSet 计算得出两个概念集的相似度,进行排序后显示出来供用户选择。

在具体实现上,使用 Jena 提供的 API 对 OWL 描述的本体模型进行操作,利用 Jena 中的方法可以获得相应概念集合来求解概念语义相似度:

①读入 OWL 描述的领域业务本体。

②使用 ontModel 的 getOntClass() 得到两个需要计算相似度的概念的 OntClass。

③使用 OntClass 的 getSuperClass 方法可得到这两个概念的超类列表。

④计算两个超类列表中超类相同的个数  $n$ 。

⑤利用上面的公式计算出两个概念的语义相似度。

## 3 基于复用的 workflow 过程定义方法

基于可复用资产进行 workflow 过程定义包括 4 个主要阶段。

(1)基于领域业务本体对具体系统进行业务分析。开发人员理解业务本体中体现的业务功能层次,并通过与用户交流将它与具体系统的业务功能需求类比,确定具体系统中需要在领域业务本体的基础上增加或去除哪些业务功能,这样就形成了具体系统的业务功能层次模型。对领域业务本体中的角色、组织单元、语义关系也进行这样的类比,以确定具体系统的角色、组织单元以及它们之间的关系。

(2)检索 workflow 模板并进行适应性修改。根据具体系统的业务需求,利用上节的检索方法在领域 workflow 模板库中寻找合适的候选 workflow 模板。根据候选 workflow 模板的描述以及模板库中的使用者评价信息,找出最适合的 workflow 模板并进行实例化,生成具体的过程定义。workflow 模板采用的是白盒复用方式,所以使用时可以根据实际情况进行修改。实例化

之后得到的过程定义也可以作为子过程,组合后形成新的过程定义。

(3)开发 workflow 模板库中不存在的过程定义。对于一些新系统比较特殊的业务功能,领域 workflow 模板库中如果不存在相应的工作流模板,需要进行开发。

(4)对可复用资产进行维护。可复用资产的积累是一个逐渐演化的过程,具体系统的过程定义对可复用资产提供反馈,根据使用者的反馈可以对领域业务本体及 workflow 模板进行修正,以不断促进可复用资产的完善和充实。

这 4 个阶段是一个迭代的、逐渐精化的过程,如图 2 所示。

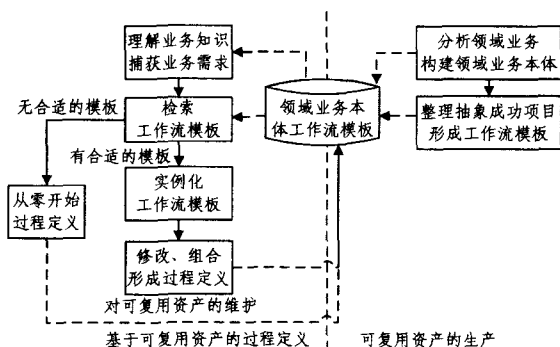


图 2 基于可复用资产的工作流过程定义方法

#### 4 基于复用的过程定义生成工具

为了验证上述方法的可行性和有效性,在扩充 SynchroFlow 过程定义工具的基础上开发了一个基于复用的工作流过程定义工具。SynchroFlow 过程定义工具主要包括:组织机构建模、过程定义、界面设计等功能模块。基于复用的过程定义工具增加了管理和使用特定领域的可复用资产的机制,主要包括:①查看 workflow 模板;②通过浏览或检索选择 workflow 模板;③细化、组合、修改 workflow 模板;④删除 workflow 模板;⑤

添加 workflow 模板;⑥实例化 workflow 模板;⑦使用情况反馈;⑧领域业务本体编辑;⑨用户管理。

**结束语** 本文在“面向流程管理的软件生产线”课题提出的架构下,研究了特定领域 workflow 过程定义的复用问题,提出了基于领域业务本体和 workflow 模板的工作流过程定义方法。该方法以已有工作为基础,充分利用过去 workflow 系统开发中积累的知识和资产,降低了过程定义的复杂度。workflow 模板等可复用资产是领域中经受过实践检验的解决方案,对它们进行复用能有效地提高过程定义的质量。基于该方法实现的过程定义工具在实际开发中的使用效果也证明了其可行性与有效性。当然,基于复用的过程定义方法依赖于可复用资产的积累与完善,整个生产线的集成环境、工具集以及对可复用资产的可信性度量也是下一步的重要研究内容。

#### 参考文献

- [1] 罗海滨,范玉顺,吴澄. 工作流技术综述[J]. 软件学报,2000,11(7):899-907
- [2] 岳晓丽,杨斌,郝克刚. 信牌驱动式工作流计算模型[J]. 计算机研究与发展,2000,37(12):1513-1519
- [3] 龚晓庆,葛玮,郝克刚. 信牌驱动模型中的工作流模式[J]. 西北大学学报,2004,34(1):13-18
- [4] 杨芙清,王千祥,梅宏,等. 基于复用的软件生产技术[J]. 中国科学(E辑),2001,31(4):363-371
- [5] van der Aalst W M P, Basetn T. Inheritance of workflow: an approach to tackling problems related to change[J]. Theoretical Computer Science, 2002, 270: 125-203
- [6] Studer R, Benjamins V R, Fensel D. Knowledge Engineering, Principles and Method[J]. Data and Knowledge Engineering, 1998, 25(1/2): 161-197
- [7] 朱礼军. 万维网环境下基于领域知识的信息资源管理模式研究[D]. 北京: 中国农业大学, 2004

(上接第 142 页)

级。这主要是因为 DEGEPO 算法不仅对基因头部和尾部进行传统进化,而且对常数域部分通过 DE 算法进行进化,提高了算法的微调性能。

**结束语** 将 DE 算法引入到 GEP 中,设计了新的基因编码和新的遗传算子。提出了一种新颖的全局函数优化算法,扩充了 GEP 的应用领域。实验验证了算法的有效性。在未来的工作中,将研究此方法在多模态函数优化中的应用。

#### 参考文献

- [1] Francois O. An evolutionay strategy for global minimization and its Markov chain analysis[J]. IEEE Transaction on Evolutionary Computation, 1998, 2(3): 77-90
- [2] 潘长城,徐晨,李国. 解全局优化问题的差分进化策略[J]. 深圳大学学报:理工版,2008,25(2):211-215
- [3] 孟红云,张小华,刘三阳. 用于约束多目标优化问题的双群体差分进化算法[J]. 计算机学报,2008,31(2):228-235

- [4] Ferreira C. Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence [M]. Second, revised and extended edition. Netherland: Springer-Verlag Berlin Heidelberg, 2006
- [5] 黄晓冬,唐常杰,李智,等. 基于基因表达式编程的函数关系发现方法[J]. 计算机科学,2003,30(增刊 A)(10):282-285
- [6] 唐常杰,彭京,张欢,等. 基于基因表达式编程发现知识的三项新技术[J]. 计算机应用,2005,25(9):1978-1981
- [7] Zhou Chi, Xiao Weimin, Tirpak T M, et al. Evolution Accurate and Compact Classification Rules with Gene Expression Programming [J]. IEEE Trans. on Evolutionary Computation, 2003, 7(6): 519-531
- [8] Rainer S, Kenneth P. Differential Evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces[J]. Global Optimization, 1997(11): 341-359
- [9] Price K V. An Introduction to Differential Evolution [J]. New Ideas in Optimization, 1999: 79-108
- [10] 王凌. 智能优化算法及其应用[M]. 北京:清华大学出版社,2001