

特定领域软件框架的提取方法研究

吴毅坚, 赵文耘

(复旦大学计算机科学与工程系, 上海 200433)

摘 要: 通过分析传统业务流程、建立简单应用框架并从简单框架泛化为领域通用软件框架, 采用诸如识别核心业务流程和周边业务流程、简化和规范化传统业务流程以及对功能单一的应用框架进行领域扩展等方法, 从设计简单而有效的软件框架入手, 通过具体的应用提取并扩展为该领域更为通用的软件框架。

关键词: 开发方法; 软件框架; 领域工程

中图分类号: TP311. 5 **文献标识码:** A **文章编号:** 0372-2112 (2003) 12A-2151-03

Research on Distilling Domain Specific Software Frameworks

WU Yi-jian, ZHAO Wen-yun

(Department of Computer Science and Engineering, Fudan University, Shanghai 200433, China)

Abstract: A method to design domain specific software framework is put forward, which is based on analysis on traditional business operations and generalization of simple application frameworks. Core operations and accessory operations are recognized in traditional business operation processes, and the operations are formalized and simplified in order to form a simple singlefunctioning application framework, which can be extended to the whole domain by comparison and generalization with other frameworks on the same core operations. The main point of this method is to build up a comparatively simple but effective application framework before a more complex and generalized domain framework can be constructed.

Key words: develop method; software framework; domain engineering

1 引言

在企业应用中, 特定领域的框架 (Framework) 是设计复用中较为具体的例子。它针对性较强, 具备一定的灵活性和可扩充性, 是一个不完备或半完备的设计复用结构。这类框架的建立, 也带动了特定领域构件技术的发展^[1]。至今, 软件的生产仍然无法像硬件那样以高度自动化的流水线方式进行。目前的软件开发现状和硬件产业的发展提示我们, 从特定领域简单的应用开始, 逐渐抽象提取出领域内统一的软件框架, 将大大提高该领域的软件开发效率。本文将探讨一种较为简捷的、面向特定领域的软件框架的设计思路和方法。

2 从特定领域传统业务流程提取领域软件框架

2.1 特定领域传统业务流程的特点

对传统业务流程的分析、简化和规范化, 是建立特定领域框架的重要一步。该方法是基于传统的业务流程有下面三个特性:

(1) 可识别性: 用户熟知传统的业务流程。尽管在与开发人员的探讨中, 相互交流有时并不顺畅, 但一些熟知业务流程

的用户将非常有助于传统业务流程的规范表述, 并为把该流程移植到软件上建立基础; (2) 规范性: 传统业务流程总是遵循一定的规章制度或者某种约定, 这种规范的流程将便于移植到计算机上, 在实际的运用中人为的随意性是可以最大限度避免的; (3) 功能单一性: 传统业务的分工是明确的, 特定的岗位, 总有特定的任务和特定的业务流程, 这些业务流程在领域内基本相同且有很大的抽象空间。

在这些特性上建立起来的软件框架将从最大限度上避免本文前一部分所说的那些困难, 可能是简单的、功能单一的, 却易于修改、扩展和组装。

2.2 从传统业务流程中提取软件框架的步骤

结合传统业务流程上述这些特性, 我们研究了多个数据库应用的设计开发过程, 认为, 设计领域软件框架应当从特定应用开发的分析设计阶段开始就做好相关的传统业务流程向计算机软件的移植工作。为此, 我们对软件生命周期中分析和设计阶段做了一些细化和扩充。这些细化和扩充, 在软件生命周期的每个迭代周期中都可能使用。

(1) 需求分析和领域分析, 建立用例图 需求分析捕获系统的总体要求。通常, 面向对象的领域分析将生成一些领域

收稿日期: 2003-10-20; 修回日期: 2003-12-26

基金项目: 上海市科学技术委员会基金 (No. 035115026); 可变粒度的软件配置管理工具研究; 国家高技术研究发展计划 (863 计划) 基金 (No. 2002AA114010); 上海构件库及其应用; 国家高技术研究发展计划 (863 计划) 基金 (No. 2001AA113070); 基于 Internet、以构件库为核心的软件开发平台; 复旦大学研究生创新基金

类^[2],这是一种自顶向下的方法;但在我们的这个方法中,由于是从业务流程中提取软件框架,是一种自底向上归纳、合并、集成的方法,因此更加侧重于业务流程的识别、提取和规范化。

(2) 识别核心业务流程和周边业务流程 所谓核心业务流程是指有可能对应用的主要数据进行增删改操作的一个或者一组用例;而周边业务流程是指对应用的主要数据只进行读取操作的用户。

如果有多个用例对应用的主要数据进行修改,那么必须仔细分析他们是否属于同一个业务流程。如果这些用例实际上是对同一组数据顺序的修改关系,那么他们应当属于同一业务流程的不同部分,应当将他们识别为一个核心业务;如果他们修改的数据不相同,那么很有可能他们属于不同的业务流程,需要区别对待。如果修改的数据都是系统的主要数据,那么就应当识别为不同的核心业务,对每个核心业务流程分别进行后面的过程;如果发现修改的数据实际上是系统的辅助数据(比如日志、状态快照等),那么应当将修改辅助数据的用例识别为周边业务流程或者融合为核心业务流程的一个组成部分,如图1示意。

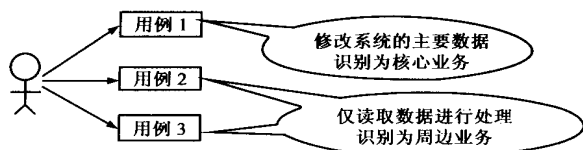


图1 识别核心业务流程

这时识别出的核心业务流程只具有功能单一性,但是细节还不明了,需要进一步细化。

(3) 参照传统业务流程,分析核心业务流程的细节 核心业务流程是从用例识别出来的,因此一开始只是一个“黑盒子”。根据传统业务流程的运作过程,我们可以把核心业务流程细化。这个过程类似于数据流图的细化过程。这个步骤的目标就是用一种用户语言,通过用户和开发者的交互,将核心业务流程的具体内容表述出来,如图2示意。

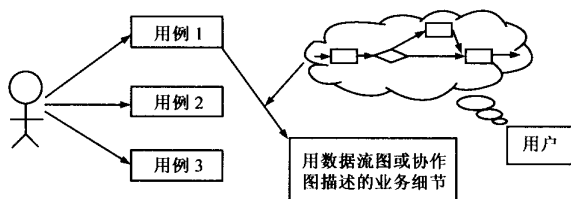


图2 描述核心业务流程

(4) 建立最初的核心业务流程模型 开发人员在与用户的交互过程中,逐渐将核心业务流程的内容用用户能够理解的形式表述出来。通常,简化的数据流图是一种非常有效的方法。简化的数据流图中可以不涉及每个操作的最细节描述,而仅仅进行1~2层的细化,直到用户和开发者都明确流程的整个过程以及各个操作之间的关系为止。这时,开发人员已经可以从这种简化的数据流图中看到核心业务流程的模型雏形。

(5) 核心业务流程模型和传统业务流程的互相适应(传统

业务流程规范化) 这是一个模型和实际流程的互动的迭代过程。

首先假定对核心业务流程模型内各个模块功能均在预期范围之内,使用一些测试用例同时对核心业务流程模型和传统业务流程进行走查。对于所有测试用例,如果核心业务流程模型能够正常运作,并且其输出与传统业务流程的输出一致,那么就可以进入(6);否则要么对当前的核心业务流程模型进行调整或者扩展,要么检查传统业务流程是否有不正确、不合理、可改进变通之处,并进行修改。通过对两套流程机制的对比检查,使得两者互相适应。这既是核心业务流程模型的完善过程,也是传统业务流程的规范化过程。

传统业务流程的变动可能影响到该业务流程的领域通用性,应当有领域专家进行分析和评估。

(6) 核心业务流程模型的框架实现(面向对象设计),并向框架补充周边业务流程接口 现在对该模型的接口用面向对象的设计原则进行编码,以形成应用的顶层框架。由于模型是以流程的各个部分为单元的,因此必须首先对该模型进行面向对象分析,识别出类,并将流程各个部分的操作封装到各个类中。这些操作仅仅是一些接口,因此不需要实现具体的细节。同时,各个操作之间的组装和调用关系如果是固定的,那么也可以在中体现这种组装和调用关系。这样,就初步完成了从业务流程模型向应用框架的转变。

(7) 为应用框架的各个接口选择具体的实现策略,完成该应用框架的一个实例 这是对应用框架的实例化,用面向对象的编程方法对应用框架的接口进行实现。这也是通常的应用开发过程。

通过上述(1)~(7),我们仅建立了一个应用框架并通过该应用框架实现了一个应用。要把这个应用框架拓展为领域框架,需要对多个同领域相似的应用进行类比和抽象。为此,还要反复地进行下面两个步骤:

(8) 识别多个应用或应用框架中隐含的可变点 所谓可变点,是指实现相同核心业务流程的不同应用或者应用框架之间的差别。这包括但不限于:接口调用方式(参数数量、顺序等)的变化;对于某些特殊数据的处理逻辑不同;核心业务流程内部接口的数量的不同;周边业务流程的个数和功能的差别。

由于这些不同的应用是针对相同核心业务流程的,因此,不同应用之间的差别通常是可以调和的。周边业务流程的不同无法合并时,需要框架提供相应的接口或将不同的周边业务流程吸收到框架中。

(9) 通过可变点的抽象化、参数化,抽象领域软件框架雏形 不同应用框架的相应可变点可以通过多种方式(稍后我们还将讨论其中一些常用的方式)进行通用化。通用化后,不同的应用框架就可以进行合并,形成完成相同功能、适用面却更加广泛的应用框架。这种向领域的扩展将逐步形成该领域的软件框架雏形。

2.3 方法的补充

上述过程是建立在传统业务流程的可识别性、规范性和功能单一性的基础上的。如果传统业务流程识别出现偏差导

致识别出来的流程功能混杂、逻辑不清,都将严重影响提取软件框架的效果。为此,需要特别的注意:

(1) 保持业务流程的功能单一性,避免不必要的功能揉合

理论上,一个成熟的特定领域软件框架的功能可以是很复杂的,但是,直接设计如此复杂的软件框架却很困难。而由相对独立、功能单一的子框架组合成更普适的框架就相对简单得多。因此进行特定领域软件框架设计和提取时,既不能将一个业务的各个部分割裂开来,也不能将几个不同的业务揉在一起,增加系统不必要的复杂度。只有当各个模块充分完善以后,才有可能通过集成的方式将它们合并起来协同工作。

(2) 识别核心业务流程和周边业务流程的标准

对于一个特定的应用而言,核心业务流程控制着应用的核心数据状态的变化,其功能通常是单一的,也相对稳定且有较强的领域通用性。

在核心业务流程周围,有多种辅助功能,如初始化、分析统计、一致性检验等周边业务。它们大部分对核心业务读写的数据库中的核心数据,通过计算、分析、加工,把处理结果提交用户界面。这些辅助功能在相同领域的不同应用中需求变化很多。在从特定应用框架中提取领域框架时应尽可能覆盖这些功能,或者为这些可变功能预留接口。图3表示了这种分解方式。

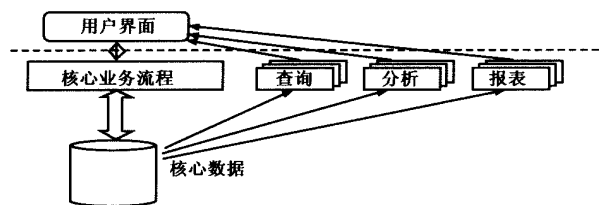


图3 业务结构识别

核心业务流程是周边业务的基础;周边业务虽然很少修改关键数据,却在整个应用中占有很大的比重。良好的设计和灵活的结构将大大提高查询、分析、统计等功能的实现效率。

(3) 传统业务规范化 在信息化过程中,业务流程不规范是将计算机软件应用于传统业务的主要障碍。在进行传统业务分析时,必须对该业务的流程予以规范化,一方面帮助用户理性的认识自己的业务流程,另一方面也明确软件的真实需求。规范化程度的高低,将决定将来框架的领域通用性的强弱。规范化实际上是使识别出的核心业务流程和传统业务流程相互适应的一个过程。

(4) 在多个系统中识别隐含的可变点及其实现 在从多个应用框架到领域软件框架的提取过程中,可变点的识别和实现是重要的一环。可变点的实现有多种策略。简单的情况可以直接用设计接口的方法来实现,而其他情况则可以根据实际的变化复杂程度采取合并、硬编码、参数化、软构件扩展槽、BNF 规则引擎等方式实现^[3]。事实上,只有经常改变的部分才需要从设计上加以抽象,形成灵活可扩充的设计^[4];而对于不经常变化或变化有限的部分,要避免过度设计。

3 结束语

我们讨论了以功能单一的核心业务流程为中心的特定领域软件框架提取方法。该方法充分利用特定领域业务流程的特性,避开直接开发需求复杂繁多的大型通用软件框架的尴尬,而直接针对传统业务流程简化、规范化,并向计算机移植。特定领域软件框架的提取和使用,反过来将促进领域内传统业务流程的规范化,减少人为的随意性,为更高层次的软件框架的产生以及更底层的特定领域构件(甚至领域通用构件)的广泛使用奠定基础。

参考文献:

- [1] 孙昌爱,金茂忠,刘超. 软件体系结构研究综述[J]. 软件学报, 2002, 13(7): 1228 - 1237.
- [2] Jacobson, 等. 统一软件开发过程[M]. 周伯生, 等, 译. 北京: 机械工业出版社, 2002. 91 - 92.
- [3] 陈兆良, 王千祥, 梅宏, 杨芙清. 面向对象领域设计中的变化性处理[J]. 电子学报, 2001, 29(1): 1488 - 1490.
- [4] Craig Larman. Protected variation: The importance of being closed[J]. IEEE Software, 2001, 91.

作者简介:



吴毅坚 男, 1979 年生于上海, 博士研究生, 主要研究领域为软件工程。

赵文耘 男, 1964 年生于江苏常熟, 教授, 博士生导师, 主要研究领域为软件工程, 软件构件技术, 电子商务。