

领域驱动的数字图书馆系统重构研究

王莉, 吕世灵, 梁冰, 白海燕
(中国科学技术信息研究所, 北京 100038)

摘要: 针对数字图书馆应用系统在运行过程中因需求变化、功能扩展导致原有设计架构混乱的问题, 提出一个基于领域驱动的系统重构方法框架, 并通过NSTL网络服务系统重构实例说明如何应用该框架开展重构工作, 重点分析采用领域驱动设计思想的建模过程。

关键词: 系统重构; 领域驱动; 数字图书馆; NSTL网络服务系统

中图分类号: G250

DOI: 10.3772/j.issn.1673-2286.2015.05.006

1 引言

数字图书馆系统建设是一项信息化工程, 涉及资源的采集、加工、组织、管理和利用等方面。随着以云计算、大数据、移动互联为代表的新型信息技术的发展和运用, 外部环境不断变迁、用户需求同样不断变化, 数字图书馆在寻求业内甚至跨界的融合与协作过程中, 不断调整业务、拓展服务, 其系统更是呈现出大规模复杂化发展的态势。随着时间的推移, 原有精心设计的良好的架构逐渐变得不堪重负, 但重新设计开发代价太高, 同时, 新建设的系统仍然会重蹈覆辙。在适当的时候对系统进行重构, 能够保持软件的设计简洁、可维护、可扩展, 保证系统的可持续发展。

2 相关知识

2.1 系统重构

系统重构是在不改变软件外部行为的前提下对内部结构进行调整的过程, 以少量修改、测试、少量修改、测试……这种“小步前进”的节奏确保整个过程安全平稳地进行。重构过程本身并不为系统增加新功能, 但是当需要变更系统时, 可以通过两个步骤完成: 首先

进行系统重构, 使之适应新的需求; 继而进行变更, 实现新的需求。这种方式又被称为“两顶帽子原则”。^[1]

按照作用层次可以将重构分为代码重构 (Refactoring)、模块重构 (Big Refactoring) 和架构重构 (Architectural Refactoring) 三大类^[2]。图1展示了研究者对三种重构的认识, 其中, 频度维标识在软件开发与维护过程中的发生频率, 能力维标识对实现者能力的要求。

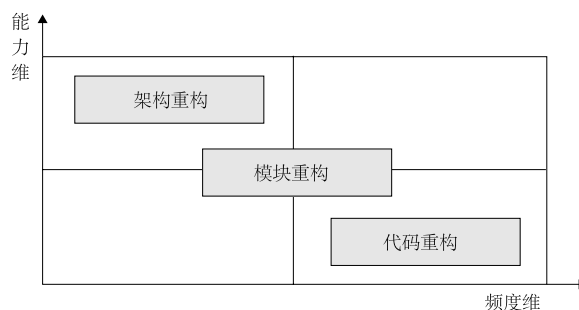


图1 三种重构的细分^[2]

代码重构发展较快, 有很多可用的技术、方法和工具, 很多主流的开发环境也都提供简单重构功能。例如Eclipse中的JDT插件支持对Java项目、类或成员进行多种类型的自动重构; 微软Visual Studio也集成了很多常见的重构功能。模块重构和架构重构表现为

设计模型的演变,并不是借助文档从抽象到抽象,而是以代码为基础,从代码级思维跳到设计级思维,抽象出当前的设计模型,继而进行重新设计,再回归代码层面验证。实际上这是一个双向工程:通过逆向工程了解系统,获得系统全景(设计“什么”),进而分析全景描述背后隐藏的设计思路(“为什么”设计);通过正向工程生成新的模型并加以实现。可见,不同层次的重构,思路和方法是不同的。面对一个软件系统,首先需要了解是否存在问题,存在什么问题,这些问题是否能通过重构来解决。确定采用重构方式之后,需要设定目标,确定重构的对象,是针对架构、模块还是局部代码,进而针对目标制订实施计划。

2.2 领域驱动设计

领域驱动设计(DDD: Domain-Driven Design)^[3,4]是2004年Eric Evans提出的一种软件开发方法,其主要思想是将领域模型作为业务分析设计的核心,并贯穿整个软件开发流程。领域模型由实体、值对象、服务、关联、聚合、仓储和工厂七种元素共同构成,其中,实体、值对象、服务和关联四种基本对象元素映射现实领域中的基本业务逻辑,聚合、工厂和仓储则维护这四种对象元素在模型中的生命运行周期,使其可以在模型内实现自身状态随时间的演变,从而使得领域模型成为一个与外界无关联,却能完整演变现实领域业务逻辑运行情况的模型对象。这样的模型是有边界的,只反映领域内所关注的业务,和任何技术实现无关,能够很好地保证现实与模型间精确而完整的映射。可见,领域模型是整个软件的核心,需要领域专家、设计、开发人员共同维护,而一个好的领域模型是控制复杂问题的关键。

领域驱动设计能够很好地指导复杂领域的软件开发,其关键在于构建一个正确的领域模型,这对人员素质要求很高;在后续软件开发过程中,进一步摒弃了传统的三层架构,提出基于自身特点的分层方式(用户界面层、应用层、领域层和基础设施层),这更是一种开发思维的转变,也是当前在实际项目中领域驱动设计没有广泛完整使用的原因。

本文将领域驱动思想引入系统重构过程,不仅是提出一套面向数字图书馆系统重构的方法框架,更希望通过实践分析其可行性,给同类应用实施提供参考。

3 领域驱动的数字图书馆系统重构方法框架

本文主要探讨数字图书馆系统的架构重构问题,对双向工程这一基本方法进行扩展,引入领域驱动思想,提出一个通用的系统重构框架,如图2所示。

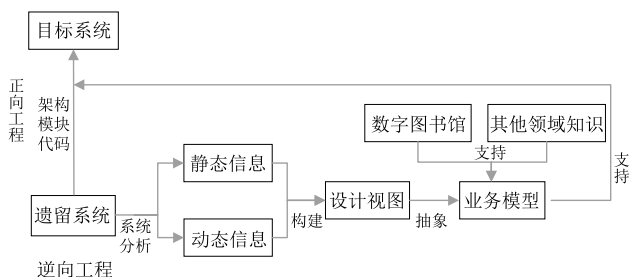


图2 数字图书馆系统重构方法框架

第一步,代码分析。从阅读源代码开始,辅以开发文档,借助各种自动化工具捕捉系统静态信息和动态信息。静态信息展现了系统的结构特征,例如不同类之间的继承、实现、依赖等关系;动态信息则描述了系统运行过程中的行为特征,例如从一种状态到另一种状态的转换。

第二步,还原设计思想。不同分析方法得到关于系统不同方面的描述,对这些描述信息进行融合形成完整视图,进而抽象获得系统高层视图。这是一个不断反复的过程,最终实现设计再现。

第三步,结构重组。运用领域知识对系统设计进行建模,产生业务模型;利用该模型指导系统重构。这一步骤集中体现了领域驱动设计思想:领域专家引入通用数字图书馆模型,结合用户实际业务和应用需求(表现为其他领域知识),会同设计和开发人员从当前系统分析中得到的设计视图,共同构造业务模型,作为项目交流的工具和语言,指导接下来的重构工作。

关于数字图书馆模型的问题,数字图书馆领域对模型的研究是伴随着数字图书馆的发展而展开并逐步演进的,呈现出两大主流研究方向。一是概念模型研究,将数字图书馆看作技术、经济、社会、法律和政策等诸多要素作用下的复杂系统,用模型的方式对数字图书馆的本质特征进行概念化和形式化描述^[5]。例如Edward Fox等人提出的5S模型^[6]、欧盟Delos工作组提出的DLRM模型^[7]和Sandor Daranyi等人提出的5M模型^[8]。二是应用模型研究,聚焦具体问题,关注软件系

统架构。例如国内外研究者在资源整合、跨库检索、个性化服务、系统互操作等方面已经提出若干系统架构,并且大都有相应的原型系统,某些已经进入实践,或者说架构本身就来自于对应用系统的经验总结^[9]。从软件工程的思想看,这两个研究方向的结合构成一套完整的模型驱动方法体系,对概念模型的研究类似于需求分析以及与平台/技术无关的系统设计,应用模型研究方向则类似于与平台/技术相关的系统设计和实现。然而,在研究中,两者缺少沟通交流,前者缺少从概念框架到应用的最佳实践,后者缺少与实践相适应的系统分析。本文提出的重构方法框架力图将两者结合起来,数字图书馆模型是指数字图书馆概念模型,以之作为整个项目实施的参考模型;设计视图则对应于应用模型;在参考模型的支持下,结合具体领域知识对当前应用模型进行修正,产生业务模型,完成对现实的抽象,同时保证这一模型经过不断迭代与精炼,是可实现的。

需要提及的是,5S模型基于计算机科学的方法论,不仅对数字图书馆体系结构进行了系统、完整的阐述,同时提供了形式化方法(包括本体关系)、基于XML的建模语言5SL、建模工具5SGraph和代码生成工具5SGen。这一系列方法与工具十分完整,体现了模型驱动思想,能够帮助用户快速建立数字图书馆原型系统。5SGen第一版就应用于MARIAN项目,支持基于语义网络的知识表达和包含行为的节点描述;第二版也在CITIDEI、VIADUCT、NDLTD Union Catalog以及BDBComp等项目/系统建立原型过程中得到应用并发展完善。文献[10]在5S理论框架基础上提出一个通用生成器,重点关注“社会”和“情景”两个纬度的建模,帮助设计人员从模型中半自动地构建数字图书馆服务,力图完成从模型到具体实现的过渡。受这一思路的启发,在接下来的案例分析中,采用了5S模型作为参考模型框架,但具体的建模并没有使用5SL、5SGraph和5SGen,而是采用了软件领域更为通用的UML语言。

4 案例分析

NSTL网络服务系统是国家科技图书文献中心面向全国科研机构和科技人员开展信息服务的核心平台,自2000年上线以来经历了三次大的改造,目前运行的是第三期系统。第三期系统建设工作启动于2007年,2010年正式上线运行,设计上采用J2EE MVC框架,将应用程序的业务逻辑、表示功能和控制功能分为模型

(Model)、视图(View)和控制(Controller)三个核心部件实现,以增加代码的复用性、可读性,减少数据描述和应用操作的可耦合度。这种设计模式有利于软件的维护和功能的扩展,有利于组件的重用。应该说,第三期建设对设计的考虑是合理的,交付的系统也具有较好的可扩展能力。然而,这种扩展能力随着越来越多新需求的加入,越来越多新代码的添加,逐渐下降,系统维护成本也越来越高。面对庞大的代码库,看到的是大量的类和类继承关系,为满足需求变化而引入的特殊处理散落在代码各处。我们需要一个更透明、更易理解的系统,以适应外部环境变化,因此,明确以系统架构重构为目标,按照上节描述的基本框架开展工作。

逆向工程部分主要是人工辅以工具完成,本节只是简述了实际工作中采用的具体步骤(4.2和4.3),而将重点放在正向工程部分,重点分析业务建模过程(4.4)。

4.1 建模语言

UML是一种定义良好、易于表达、功能强大并且普遍适用的建模语言,选择其2.0版本作为团队内部交流的工具,通过UML图描述并传达设计思想。需要注意的是,设计思想的表达是关键,因此不能为了画图而淡化思想表达这一根本目的。在项目初期,更多的是采用一些更直接的、更贴近人类思维方式的手段(例如演讲、文字和简单图形)共享知识、推敲模型。当这些需求基本确定之后,再转换为正式的UML图并固化下来,保证团队工作在一个清晰的、认知一致的模型下。

辅助建模工具采用Rational Rose V7.0。

4.2 代码分析

代码分析主要包括三个步骤。

步骤一,整理文档,获取系统基本设计。系统建设期间保存了完整的设计文档,主要包括需求分析、概要设计、详细设计和数据库设计,构成对系统的自顶向下的综合描述,这是团队工作的起点。但是,在5年运行期中,不断进行着程序纠错和功能增强等工作,原有结构已然破坏,部分高层设计随着时间而丢失或过时,加之开发人员变化,设计文档的维护工作缺乏监督,很容易造成文档和代码之间的不一致。因此,设计文档展现的结构只能作为基础,还需要通过对源代码自底向上的分析对其进行修正。

步骤二, 分析数据库中物理结构, 提取数据模型。第三期系统采用关系数据库, 物理结构主要是指表的结构、表的定义、表的列定义, 以及表之间的约束关系(主键和外键)。在这一步骤中需要对关系模式进行识别, 将逻辑模式转换为概念模式, 并用面向对象的方式重现。从关系数据库的物理结构中获得概念模式, 已经有一些通用的方法^[11], 但是这些方法对数据库范式提出不同要求, 在本项目中并不适用。为了提高数据库的查询效率, 第三期系统数据库结构中引入了若干冗余和非规范化设计, 例如, 全文文献订单表中不仅包含文献唯一标识符(外键, 关联文献信息表), 还包含了文献题名、语种、作者、出处等更详细的信息, 可以直接支持业务统计, 而无需执行表间关联。这种设计是面向现实需求且适应工程应用的, 但也使得数据库结构表现出来的语义变得模糊。项目实施过程中对数据库结构的分析主要采用人工方式进行。

数据是应用中最稳定的部分, 通过对数据库的分析可以得到大部分的实体关系, 数据库逆向先行的做法可以大大减少问题的复杂程度。但是, 数据库模型并不能表达概念模型所有的结构和约束, 理想情况下, 这些丢失的结构可以在过程部分得以体现, 因而也可以通过源代码分析的方法发现。

步骤三, 采用人工为主工具为辅的方法解读源代码。以步骤一得到的整体视图为基准, 对第三期系统进行粗粒度划分, 在划分范围内考察类、对象、方法以及它们的依赖关系, 重点分析代码中的分支、循环、调用, 以及变量之间的联系, 捕获隐藏的业务逻辑。这一步骤非常繁琐, 也是基础性的、无法跨越的, 具体操作中和局部代码重构工作相结合, 在解读过程中主要针对代码可读性问题, 一点点重构, 一点点测试。底层细节的积累将使得系统整体概念趋于完整。

4.3 还原设计视图

通过代码分析会获得一组UML图, 这些图是通往模型的窗口, 特定的图会显示模型的某些部分。项目组采用RUP4+1视图方法^[12], 将这些图按照逻辑视图(Logical View)、过程视图(Process View)、实现视图(Implementation View)和部署视图(Deployment View)分类组织, 通过一组用例图(User-Case View)对重要场景进行说明, 最终形成系统的完整架构。图3展示了系统高层最基本的用例图。

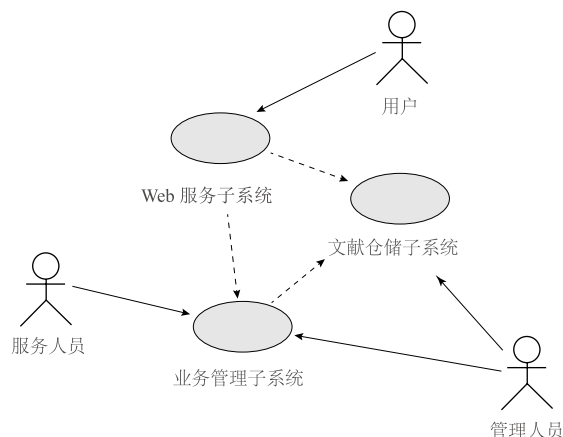


图3 NSTL网络服务系统基本用例图

4.4 业务建模

以5S模型为参考模型框架, 结合NSTL网络服务系统实际需求, 对当前的应用模型进行修正, 产生业务模型。这一过程中强调以业务过程为中心、目标为导向, 重点在于对核心业务过程的捕捉与描述, 而无需关心这些业务过程是否真正需要自动化。

1) 5S分析

5S理论从数据流(Stream)、结构(Structure)、空间(Space)、情境(Scenario)和社会(Society)五个要素定义数字图书馆, 对照这五个要素对NSTL网络服务系统(以下简称NSTL系统)进行分析, 主要明确以下问题: A.系统支持哪些媒体信息(数据流); B.这些信息如何组织起来(结构); C.组件之间如何关联, 以什么方式呈现给用户(空间); D.系统行为(情景); E.服务的使用者和管理者(社会)。

首先, NSTL系统中最基本的数据流是文本和图像, 采用序列化方式存储读取, 存在GBK和UTF8两种编码格式。原因在于, 系统建设之初, GBK是业界主流编码格式, 因此原始代码、文献数据文件、文献仓储都是基于GBK建立的; 随着UTF8的兴起, 相关技术越来越成熟, 文献数据在进行内容格式调整的同时率先更改为UTF8编码格式, Oracle文献仓储随之进行数据清洗, 采用了UTF8编码, 直接相关的模块(数据质检与加载)也进行了代码重构。然而, 仍然有采用GBK编码格式的原始代码存在, 导致在极端情况下, 页面显示出现乱码。

其次, 结构是对资源内容组织特性的描述, 主要包括数字对象内部结构、资源描述元数据规范和知识组

织工具三类。数字对象是表示信息的基本逻辑单位。以NSTL购买的Springer回溯数据库为例,一本丛书作为一个数字对象保存,其组织结构形如:丛书系列号→丛书号→丛书组成部分号→篇章号→章节元数据描述文件→章节内容文件。在资源描述方面,NSTL采用XML Schema定义了一套完整的元数据规范,覆盖期刊、会议录、学位论文、科技报告、专利和标准文献等多种资源类型,支持数据交换。在知识组织方面采用的工具主要有NSTL资源分类规范、中图法、IPC国际专利分类表、STKOS科技词表、会议规范文档和NSTL资源本体。

空间是由一组对象以及施加于这些对象之上的操作所构成的,主要表现为基于结构要素建立的索引(TRIP全文索引、SOLR全文索引)以及用户界面(基于Neo4j图形数据库的可视化)。

情境是从用户角度描述系统的外部行为,即展示特定用户群体如何从系统中获得相应的服务。NSTL系统中主要的情境有检索、浏览、订购、代查代借、订单处理、数据收割、数据质检与加载等,可分为文献仓储管理、用户服务、用户管理、文献服务和账务管理五大类。

社会是所有实体和关系的集合体,包含使用系统的人以及支持系统运行的软硬件。NSTL系统用户群体可分为外部用户和后台用户两大类,外部用户是指那些使用或调用NSTL对外服务的个人、机构和应用系统;后台用户按照角色可以分为文献仓储管理员、文献传递服务人员、用户管理员、账务管理员和系统管理员五类。

通过5S分析掌握系统全貌,得到系统内部主要的实体类和它们之间的基本关系。图4-a给出了用户、索引

文件、目录、文献之间的基本关系,突出了搜索、浏览、订购、索引、绑定五大关联;图4-b从后台文献传递服务的角度展示了服务人员、订单、用户、文献、花费之间的关系,突出订单处理环节。

2) 流程梳理

重点关注5S分析中社会 and 情境两个要素,结合实际应用需求,采用领域驱动设计方法,对逆向工程中得到的各种设计视图进行修正,以迭代方式进行业务建模。这是一个针对领域服务的建模过程,强调以领域驱动设计方法为指导,抓住上下文、精炼和大比例结构三大应用原则,尤其是大比例结构划分原则,帮助我们将注意力集中在核心业务上,抓住设计的重点。通过对核心业务的梳理,进一步澄清存在的问题。

NSTL系统目前有文献仓储、业务管理和Web服务三大子系统(见图3)。文献仓储主要完成元数据收割、质检、保存和发布。对于NSTL自加工数据来说,从数据格式规范到数据流转整个业务流程经过磨合,是相对稳定的,不存在大的问题。随着数据来源渠道的不断扩展,更多需要考虑的是将不同来源、不同格式的元数据集存放,采用深加工工具进行知识挖掘、合理关联,构建知识关联网络。这实际上是引入了新的业务流程,就重构工作而言,需要考虑的是现有系统架构是否能够支持新的业务,适当引入NoSQL产品,重构工作的重点是数据库的迁移。业务管理包括文献服务、用户审核、账务管理和系统管理等。Web服务以文献检索与全文传递为核心,还包括代查代借、参考咨询、用户注册、自助中心等。这两个子系统内部的业务逻辑都比较复杂,同时,子系统之间还存在不同程度的交互,尤其是前台下单

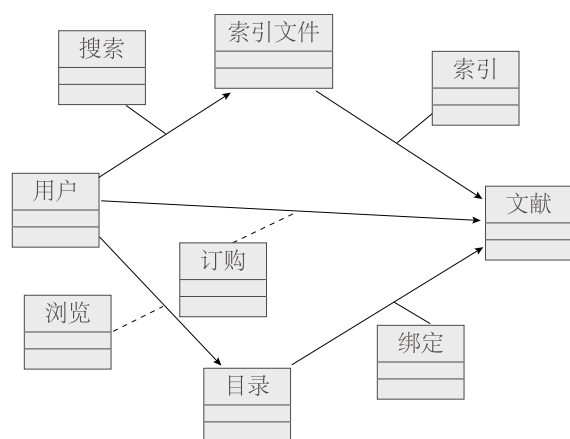


图4-a 用户-文献

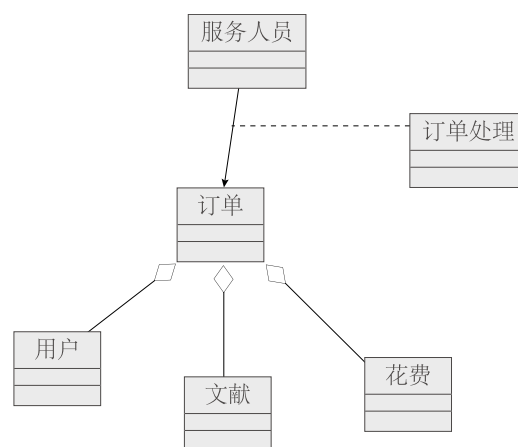


图4-b 服务人员-订单

图4 主要实体类及其基本关系

与后台订单处理各种状态交互纠缠在一起,是整个系统重构的“重灾区”。

示例1:厘清边界——全文传递订单业务流程分析

当前系统下订单的业务逻辑是从购物车管理开始的:用户选择一篇或多篇文献加入购物车;允许删除购物车中的文献;查看购物车,确定选择,填写订单信息并提交;计费;扣款;生成有效订单。下订单操作完成之后,用户可以对订单进行管理。这个过程看似简单流畅,然而随着越来越多需求的加入逐渐变得复杂,例如,为保护知识产权,应外商要求增加订购限制条件“每人每天订购同一本外文期刊或同一本外文会议录中的文献不能超过三篇”;计费规则变化,简单区分文献类型的收费标准不再适用,需要对特定文献集合制定特定的收费标准,同时,用户类型也影响计费;扣款规则变化,适应中心服务的推广策略,用户拥有试用卡、充值卡、集团用户分配、普通预付款等多个“钱包”,在扣款环节涉及“扣款顺序”和“合并支付”问题,同样,退款也存在相应的问题。这些变化导致原本清晰的类变得庞大臃肿,方法变得越来越长,传递的参数也越来越多。

图5采用序列图的形式重现这一过程,可以很自然地得到购物车、订单、计费 and 用户帐号等4个领域类,而窗口类则和领域类分离,用用户界面管理表示。

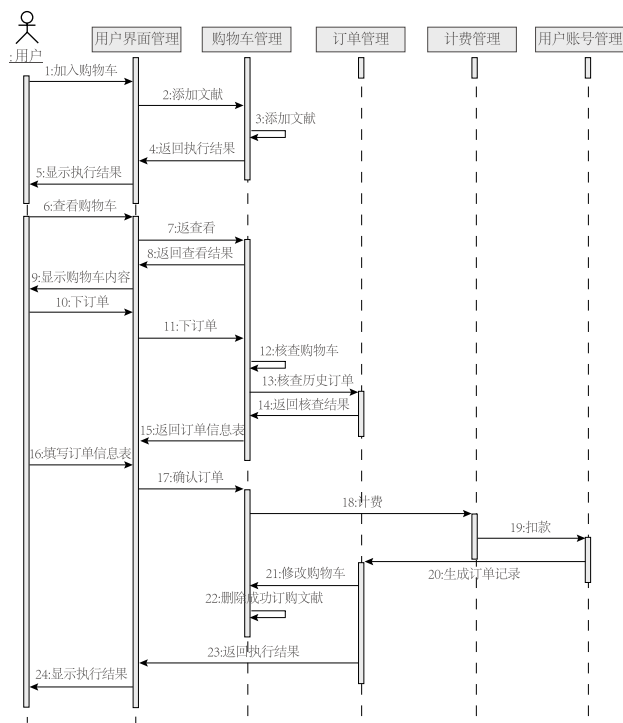


图5 全文传递下订单业务序列表

示例2:澄清问题——全文传递订单处理业务流程分析

全文传递订单处理流程可简单地概括为:订单接收;处理中;处理完成。NSTL全文传递服务是基于印本资源开展的,因此订单处理大多为线下工作,线上通过订单状态的变化反映工作进展,如图6所示。

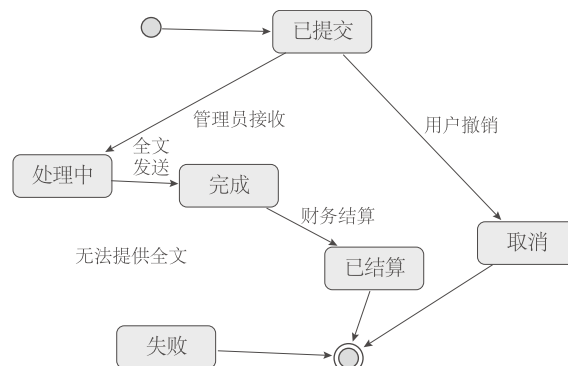


图6 订单状态变化

订单处理有两个非常重要的环节,一是核定订单费用;二是给全文电子文件加数字水印。在进行费用核定时,如果实际费用低于系统预扣费用,需要执行退费操作,通过与用户帐号类的交互完成。给全文加数字水印是后续增加的功能,由独立的数字水印模块完成。当采用电子邮件传递方式时,“扫描-加水印(数字水印管理客户端)-发送”这一过程都是脱离NSTL主系统完成的,对于订单处理人员来说,只是在系统中修改订单状态。但是对于缓存区下载这种全文获取方式而言,服务人员线下扫描,在系统界面中选择扫描文件,通过上传操作“一步”完成加水印(接口调用)和上传工作,两个任务绑定在一起,相互影响,经常造成整个上传操作失败。

4.5 重构

在业务分析基础上,提出新的系统架构,如图7所示。

- 主体分为四大部分:“仓储”负责系统的资源存储和服务,“子系统管理+系统日志”负责系统的底层支持和日常维护,“应用+管理”负责实现系统的主要业务流程,“展示”负责功能集成、用户交互等;

- 仓储子系统部分,引入目前已较为成熟的BigTable、列存储、存储分片等技术,支撑对NSTL的文献资源的深度梳理;

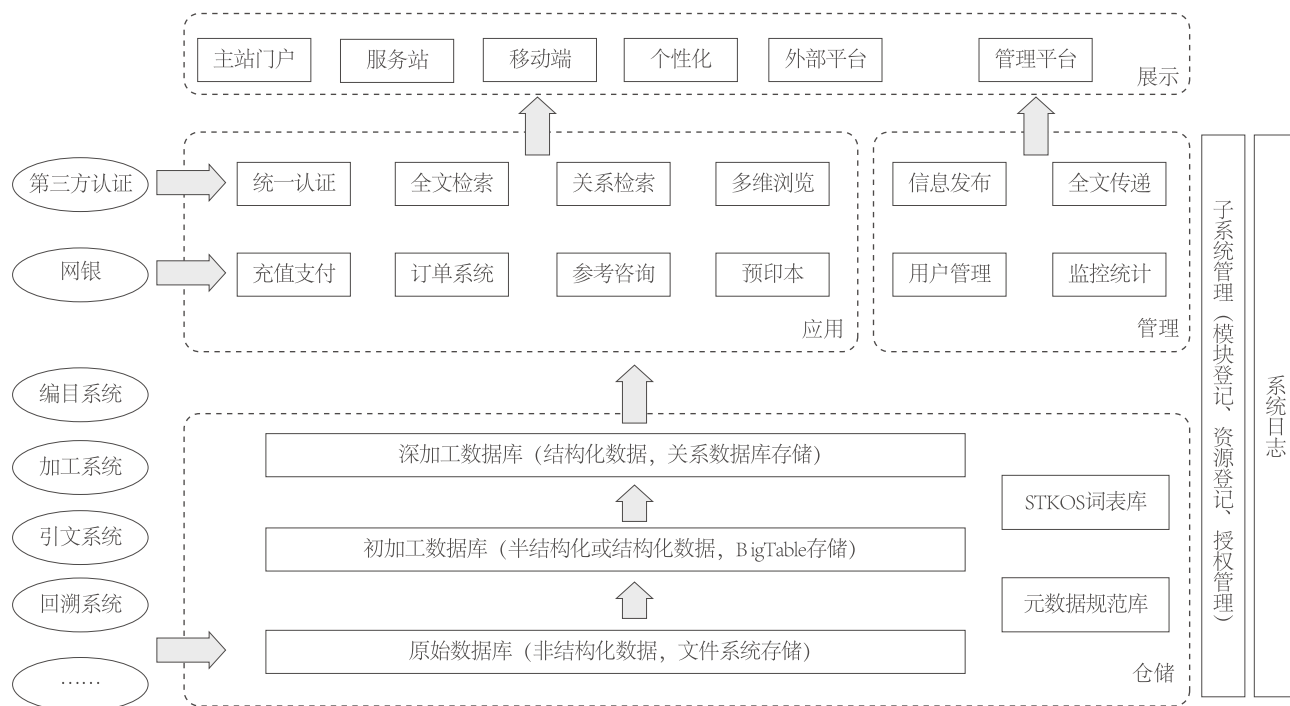


图7 新的系统架构

● 基于登记和日志机制，实现整个NSTL系统的管理和维护，将这部分功能独立划分出来，和业务流程互不干扰；

● 在前端服务中引入集群设计，提升系统的稳定运行能力，并能根据需要横向扩展；

● 建立单独的认证授权子系统，为NSTL平台的各个子系统提供统一的权限管理，同时增加对第三方认证的支持。

需要注意的是，这一架构是对现有系统架构的优化，包含了以往可重用的部分以及计划添加、修改或替换的部分。这也体现了重构的“两顶帽子原则”。

5 小结

本文将领域驱动思想引入系统重构过程，提出一套面向数字图书馆系统重构的方法框架，首先通过逆向工程捕获当前系统代码的功能及关系、系统架构、主线业务情景等重要信息，全面掌握系统架构基线；进而运用领域知识对系统设计进行建模，产生业务模型；利用该模型指导系统重构。

该框架具有以下特点：1) 以5S模型作为工程实施的参考模型；2) 吸收当前数据库和软件工程领域领先的技术方法。本文在该方法框架的指导下对NSTL三期

系统的重构工作进行分析，希望能给同类工程实施提供参考。

参考文献

- [1] 范钢. 大话重构[M]. 北京: 人民邮电出版社, 2014.
- [2] 温昱. 架构重构趋势谈[J]. 程序员, 2011(1): 47-49. <http://www.programmer.com.cn/4555/>.
- [3] Eric E. Domain-Driven Design-Tackling Complexity in the Heart of Software[M]. Addison-Wesley Professional, 2004.
- [4] 池建强. 领域驱动设计和实践[EB/OL]. <http://www.infoq.com/cn/articles/cjq-ddd/>.
- [5] 汪永红, 龚立群. 国外数字图书馆模型的分析、演进和比较[J]. 图书馆论坛, 2013, 33(2): 26-32.
- [6] Goncalves M A, Fox E A., Watson L T, et al. Streams, Structures, Spaces, Scenarios, Societies (5S): A formal model for digital libraries [J], ACM Transactions on Information Systems (TOIS), 2001, 22(2): 270-312.
- [7] Candela L, Castelli D, Pagano P, et al. Setting the Foundations of Digital Libraries: The DELOS Manifesto[J]. D-Lib Magazine, 2007, 13(3-4).
- [8] Darányi S. Toward a 5M model of digital libraries[C]. Digital Libraries for International Development Workshop in conjunction

- with International Conference on Asian Digital Libraries, 2010: 3-13. 2003:263-275.
- [9] 侯三军, 严明. 数字图书馆系统架构研究评述[J]. 数字图书馆论坛, 2011(9):19-30. [11] 王玉花. 数据库逆向工程——函数以来的挖掘[D]. 北京: 北京交通大学软件工程, 2006.
- [10] Rohit Kelapure, Marcos Andre Goncalves, Edward A. Fox. Scenario-based generation of digital library services[C]. Proceedings of the 7th European Conference on Digital Libraries (ECDL 2003), [12] Kruchten P. The 4+1 View Model of Software Architecture[J]. IEEE Software, 1995, 12(6): 42-50.

作者简介

王莉, 女, 硕士, 研究员, E-mail: wangli@istic.ac.cn。
吕世灵, 男, 硕士, 工程师, E-mail: lvsj@istic.ac.cn。
梁冰, 男, 博士, 高级工程师, E-mail: liangb@istic.ac.cn。
白海燕, 女, 硕士, 研究员, E-mail: bhy@istic.ac.cn。

Research on Digital Library System Reconstructing Based on Domain Driven

WANG Li, LV ShiJiong, LIANG Bing, BAI HaiYan
(Institute of Scientific & Technical Information of China, Beijing 100038, China)

Abstract: In order to solve the architecture confusion problems of digital library systems caused by demand changes and function expansion, this paper proposes a system reconstructing method frame based on domain driven. A case study, NSTL network service system, is provided to exemplify how to use this method frame to carry out the reconstructing work, and the modeling process based on domain-driven is empathized.

Keywords: System reconstructing; Domain driven; Digital library; NSTL network service system

(收稿日期: 2015-04-05; 编辑: 刘伟)

■ 书 讯 ■

《科技报告体系构建研究》

为推进我国科技报告制度建设, 强化科技报告资源共享服务, 贺德方研究员率领中国科学技术信息研究所科技报告研究团队, 进行了国家社会科学基金重点项目“中国科技报告资源体系构建”(11ATQ006)研究, 并对20多年来中国科学技术信息研究所相关研究和实践进行了归纳、凝练、整理和补充, 最终形成了《科技报告体系构建研究》一书。

本书作为国家社会科学基金重点项目的研究成果, 总结了科技报告产生发展的管理历程、凝练了科技报告制度的建设路径、制订了科技报告资源的整合方案, 提出了科技报告体系的构建模式, 归纳了科技报告实践的操作过程。本书对各级科技计划管理人员强化科技计划项目过程管理具有借鉴作用, 对科研人员撰写高质量科技报告具有指导作用, 对各类科研机构做好科技报告呈交、推进科技项目的规范管理和机构知识库建设具有参考价值, 对图书信息机构做好科技报告深层次加工和收藏利用具有引导作用, 也可供高校信息管理、科技政策与管理等专业研究生学习参考。

《科技报告体系构建研究》于2014年12月由科学技术文献出版社出版, 定价78.00元。