

面向信息服务的界面原型驱动开发模型

荣辉桂^{1,2}, 陈旭毅¹, 张向荣¹

(1. 武汉大学信息资源研究中心, 武汉 430072; 2. 湖南大学软件学院, 长沙 410082)

摘要: 面向服务架构(SOA)的开发和管理使信息管理系统可以实现灵活的业务流程, 根据 SOA 的优点提出一种面向信息服务的界面原型驱动开发模型, 给出相应的形式化描述。基于该模型, 结合信息管理项目实例, 分析如何应用该模型指导项目开发过程, 较好地实现从需求分析到有效设计的快速切换。

关键词: 面向服务架构; 信息服务; 界面原型驱动开发; 敏捷过程

Information Service-oriented Interface Prototype Driven Development Model

RONG Hui-gui^{1,2}, CHEN Xu-yi¹, ZHANG Xiang-rong¹

(1. Research Center of Information Resource, Wuhan University, Wuhan 430072;

2. School of Software, Hunan University, Changsha 410082)

【Abstract】 Service-Oriented Architecture(SOA) can provide flexible business flows for information management system. According to the merits of SOA, this paper designs and constructs an Interface Prototype Driven Development(IPDD) model oriented to information service and presents some formal descriptions of IPDD model, where four sub-processes are discussed respectively. An instance is analyzed to show how to apply this model to guide the software development process to accomplish fast conversion from requirements analysis to effective design.

【Key words】 Service-Oriented Architecture(SOA); information service; Interface Prototype Driven Development(IPDD); agile process

1 概述

目前 Web 服务正在成为一种基于标准技术的计算平台, 它支持构建松散耦合的广域分布式系统, 在软件工程中已经得到广泛的应用。面向服务架构(Service-Oriented Architecture, SOA)是基于 Web 服务的一种抽象的、松散耦合的粗粒度软件架构^[1]。信息服务是指将反映个体特征的相关信息或信息需求相组合, 以服务的方式予以组织和实现, 并基于一定的架构提供给服务的请求者或服务消费者的过程和模式。信息服务的组织架构本质上是基于面向服务架构的, 更强调以信息特征和需求特性实现和组织服务。

本文提出并构建一个面向信息服务的界面原型驱动开发(Interface Prototype-Driven Development, IPDD)模型, 重点设计了需求分析、面向信息服务的功能描述、界面原型建模和面向信息服务的界面原型驱动开发 4 个子过程。该模型面向信息服务构建, 符合 SOA 的技术规范, 大大提高了适用能力, 较好地支持从需求到设计过程的快速切换。

2 面向信息服务的IPDD模型

Perry 和 Wolf 根据软件生命周期各阶段响应的实体、属性、关系、主要产品和评估标准, 将软件开发过程分为需求分析、体系结构设计、详细设计和实现 4 个阶段。随着面向服务开发(Service-oriented development)思想纳入到软件生命周期中, IT 技术开发的复杂性和 IT 成本进一步降低, 因此, 开发和部署中采用 SOA 架构更有利于创建各种新的战略方案^[2]。

为突出“敏捷”、“面向信息服务”、“界面原型”3 个因素在需求向设计快速过渡中的作用, 将软件过程重新划分为需

求分析、面向信息服务的功能描述、界面原型建模、界面原型驱动开发 4 个过程, 提出面向信息服务的 IPDD 概念, 并建立 IPDD 模型; 应用函数理论和过程控制方法确保 IPDD 过程的设计结果满足完备性、一致性和正确性的要求, 使 IPDD 模型符合工程原则。

基于 IPDD 模型形式化定义以及过程划分, 引入迭代的概念, 设计面向信息服务的 IPDD 模型, 如图 1 所示。

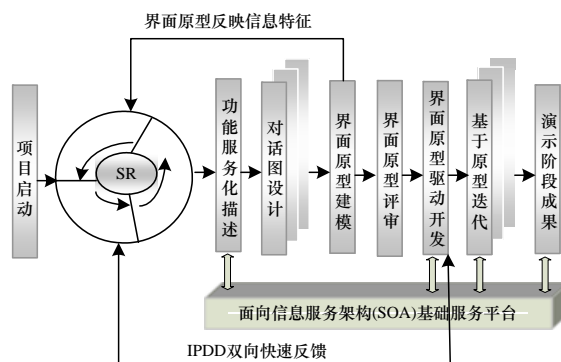


图 1 面向信息服务的 IPDD 模型

面向信息服务的 IPDD 模型由需求分析、面向信息服务的功能描述、界面原型建模、界面原型驱动设计 4 个过程组

基金项目: 国家自然科学基金资助项目(60673093); 湖南省自然科学基金资助项目(07JJ6137)

作者简介: 荣辉桂(1976—), 男, 讲师、博士研究生, 主研方向: 信息系统, 服务工作流, 电子商务; 陈旭毅、张向荣, 博士研究生

收稿日期: 2009-08-29 **E-mail:** ronghg@sina.com

成, 每个过程又分为相关子过程, 这是 IPDD 模型所定义的核心内容。

(1)需求分析。根据需求的分层结构, 设计一个面向信息服务的需求层次模型, 进行形式化定义和分析。依据文献[3]定义的功能需求获取过程模型, 进行功能需求获取分析。

(2)面向信息服务的功能描述。从需求分析阶段获得功能需求的条目, 基于 WSDL 进行服务化描述。

(3)界面原型建模。应用 UML 用例建模过程分析典型用例并设计特定用例对话框, 基于对话框进行界面原型建模, 快速构建界面原型。

(4)面向信息服务的界面原型驱动开发。以界面原型为基础, 定义驱动开发过程, 指导基于功能的服务迭代开发。

IPDD 模型面向服务架构, 定义了从需求到设计的一些规范过程, 图 1 展示了 IPDD 模型定义的各个过程及其关系。黑色实心箭头表明沿该模型的分析、执行方向及从需求分析直至最终展示阶段开发的成果。下方的双向箭头表明在界面原型驱动开发时, 应该不断与前面的需求分析结果进行对照, 确保开发的成果符合信息特征和需求特性。而在界面原型驱动开发阶段可能找到前述需求分析错误之处, 则需要尽快调整前面的分析成果, 因此, 双向箭头表明 IPDD 过程, 与需求分析阶段进行双向快速反馈。上方的单向箭头指明界面原型建模应该以用户需求中的用例为基础, 保证界面原型是用户所期望的系统风格和式样。

应用 IPDD 模型指导开发过程, 可以使开发者专注于业务问题本身, 使服务接口与执行技术明确分离, 从而对不断变化的业务需求作出更快的响应。

3 需求分析和服 务化描述

3.1 需求分层模型

软件项目要取得成功, 首先要明确所开发软件的需求, 即需求分析要解决的问题。文献[4]论述了常规软件过程的需求层次理论, 并设计面向信息服务的需求层次模型。

3.2 功能需求条目获取

基于面向信息服务的需求层次模型, 进行功能需求获取过程分析, 这是后续界面原型建模的基础。脱离了需求分析, 后续过程将无法进行。

功能需求是规定开发人员必须在产品中实现的软件功能, 它描述了开发人员需要实现的具体内容。而产品信息特性或者特征(feature)是指一组逻辑上相关的功能需求, 它们为用户提供某项功能, 使业务目标得以实现^[5]。

文献[3]设计了需求功能条目的获取过程模型, 本文在其基础上提出 IPDD 模型并予以简单介绍。

从面向信息服务的需求层次模型分析可知, 下层的每个需求要素都从上层继承而来, 且仅仅是上层的一个子集。因此, 一个业务目标可能对应多个用例, 一个用例对应多个功能需求, 一个功能需求条目对应多个界面元素。基于功能需求获取过程模型, 从“Cyber 网络书社”项目的业务需求目标入手, 分析核心用例, 抽取功能需求条目, 获取“请求某一书籍”这一用例 UR-1 相关的功能需求 FR-1 描述如下: 如果书社库存中包含所请求的书籍, 系统则显示库存中含该书名的所有书籍列表; 用户从列表中选择指定的图书下订单, 如果库存无该书, 则可向外部出版社订购该书。

3.3 功能信息的服务化描述

从业务的角度来看, 服务(service)是对应于真实的业务活动或可识别的业务功能的 IT 资产。对功能需求进行分析和获

取之后, 应按合适的粒度对功能需求条目进行分解或归并, 以便一个功能需求类能较好地映射为一个信息特征。然后基于该信息特征进行服务的形式化描述, 作为界面原型的一个底层组件接口(目前还只是一个结构, 未实现功能需求), 确保在后期的驱动开发过程中, 作为 SOA 架构的一个服务组件, 较好地支持面向服务应用。

WSDL 是基于标准的、可扩展的、基于 XML Schema 的一种理想的服务定义语言, 它描述了一个 Web 服务的各项基本信息, 如服务名称(服务在 UDDI 中注册的服务名)、所提供的操作和相应输入和输出。因此, 从 WSDL 层次描述服务是一个可行的方法, 通过识别一组功能, 定义该功能基于 Web 服务的描述语义。

定义 根据 WSDL 的 Schema, 一个 Web 服务可以标识成一个五元组:

$WS = \langle \text{ServiceName}, \text{Op}, \text{Input}, \text{Output}, \text{Profile} \rangle$

其中, ServiceName 是该 Web 服务的名称; Op, Input, Output 和 Profile 分别定义如下:

(1)操作(Op): WSDL 中每个操作 Op 对应一项功能 f (如 getBookname), 标识为一个四元组 $\langle \text{name}, \text{input}, \text{output}, \text{portType} \rangle$, 其中, name 表示操作名; input/output 表示该操作的输入/输出参数; portType 为提供该操作的接口。一个 Web 服务所能对外提供的操作为所有操作的集合。

(2)输入(Input): 每个输入标识为一个二元组 $i = \langle \text{sp}, \text{Op} \rangle$, 其中, sp 是该输入的参数属性(参数名和类型); Op 是包含该输入的操作。一个 Web 服务的输入/输出为其所有操作输入/输出的集合。

(3)输出(Output): 定义与输入类似。

(4)服务概要(Profile): 描述服务的所有黄页和白页信息, 如服务在 UDDI 中的注册 ID、服务提供者 ID、服务提供者名称、商业领域分类(domainkey)。

基于上述分析获取的功能需求类 FR-1, 以 Web 服务的形式语义, 可分析得到其服务接口描述结构:

$WS = \langle \text{ServiceName}, \text{Op}, \text{Input}, \text{Output}, \text{Profile} \rangle$

其中, ServiceName 描述为“请求某本书”; Op 描述为请求者输入书籍名称或作者, 系统则从书社数据库中按给请求者一本新书, 或者让他提交一个从外部出版社订购的请求, 从而满足其购书要求; Input: 某本书名或者编号; Output: 指定书籍订单成功并打印; Profile: 说明该服务的前置条件等。

通过需求后期的服务化接口描述, 上述相对独立的功能需求条目在驱动开发过程中被实现为一个服务组件。需求分析过程中的功能需求条目“服务化”将使未来的系统进一步适应业务需求的变化, 更好地满足 SOA 架构的要求, 从而快捷地构建灵活的系统。

4 面向信息服务的 IPDD 驱动开发过程

IPDD 模型展示了从需求到设计的快速切换过程, 该过程以界面原型驱动后续开发, 其核心是功能需求获取和界面原型的建模过程。IPDD 模型借鉴 UML 中对话框(dialog map)模型进行一系列外界与系统之间的对话元素分析, 根据对话框的界面状态转换决定界面的元素及相关链接和导航, 这也是界面原型建模的基本原理。

4.1 对话框描述和设计

对话框将用户界面视为一个有限状态机, 在某一个时刻只有一个对话元素可以接受用户输入。在激活的输入区中, 用户可以根据自己所采取的动作导航到某些其他的对话元

素。在一个较复杂的图形用户界面中，可能的导航路径数目会很大，其选项通常也是可知的。对话图及其状态转换反映了用户需求界面元素内容和元素间导航的路径，为界面原型建模提供了依据。以文献[6]中对话状态的描述语义为基础，根据对话图建模过程，得到基于前述用例 UC-1 的对话图 DM-1，如图 2 所示。

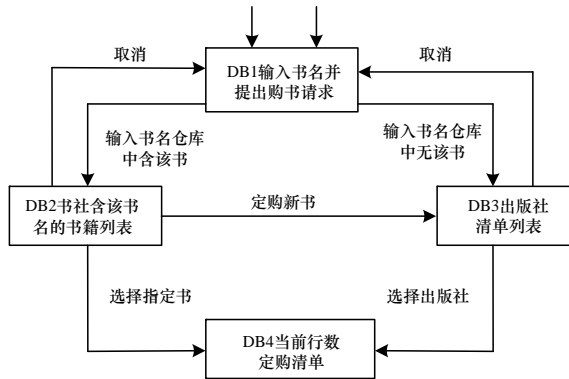


图 2 “请求某本书”用例对话图 DM-1

图 2 中的矩形框表示对话图元，这是设计界面原型的重要素材，箭头则表示从一个对话图元切换到另一个对话图元的可能导航路径。对话图 DM-1 很好地描述了用例中外部执行者通过对话图元与系统进行交互的过程，它从较高抽象层次上表示用户界面的设计，并展示了系统的对话元素及这些元素之间的导航连接，连接之间的切换通过执行用例中描述的功能动作完成。因此，基于用例的对话图建模通常用作需求图形化和界面原型建模的成果。

IPDD 过程的界面建模主要依据前述的需求分析对功能进行抽取，以“简化”的对话图模型为基础，快速探索未来用户界面原型。用户和开发方以对话图为基础，进一步描述用户可能与系统进行交互的途径。

4.2 界面原型建模实例

依据文献[6]的界面形式化描述，初始状态 q_0 是人机图形界面环境运行时进入的第 1 个界面状态，因此，在实例中首先设计主页面的界面原型。主界面原型上通常排列通用的功能列表，用一个 LOG 表现，当导航到其他页面时，这些 LOG 基本不变。

通过前述分析，基于 DM-1 对话图进行界面原型设计，取 DM-1 中的对话图元 DB-1，结合相关用例和功能需求，快速构建出基于 DB-1 对话元素的界面原型草图 IP-1，如图 3 所示。

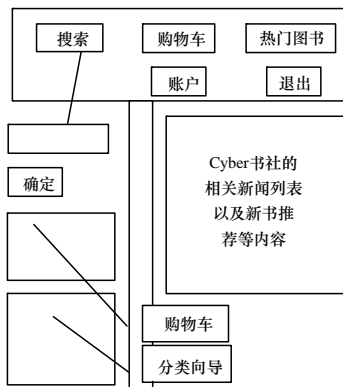


图 3 界面原型 IP-1

在 DM-1 对话图中，通过 DB-1 对应的界面草图，输入

要找的书名，基于对话图的导航路径，就应该导航到 DB-2 对话图元上。结合 DB-2 对话图元及与之相关的系统需求，快速构建界面草图 IP-2，从界面原型 IP-1 切换到 IP-2，即反映这种对话路径的执行结果，如图 4 所示。

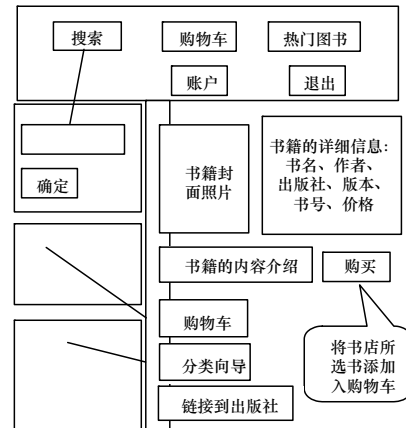


图 4 界面原型 IP-2

当 DB-1 导航到 DB-2 时，DB-2 对话元素有 2 种可能的导航模式：直接选中指定书籍订购该书，或者链接到出版社页面，向出版社订购该书。因此，针对 DB-2 的对话图元，其界面原型应该包括导航到 DB-3 的导航路径，即通过“链接到出版社”的界面元素来体现。然后，基于 DM-1 的其他对话图元的界面原型依次快速构建。而在原型设计的过程中，要注意考虑界面切换的响应动作以及其他的系统要求。

同时，界面原型设计需要与客户深入讨论、经过 2 次~3 次原型评审才能定稿，讨论时应该有最终用户类代表参与，并充分沟通，尽早获得反馈，及时在原型草图中更正。如发现需求理解上的不一致，直接修改界面原型或重画界面草图即可，修改的成本非常低廉。下一步工作则由美工根据界面草图制作出真正的页面，而不仅仅是界面原型草图。

4.3 面向信息服务的界面原型驱动开发

用户界面与内部处理模块既相互独立，又有非常密切的联系，即用户界面与内部的逻辑功能模块是相对独立的，但功能的实现又依赖于界面的输入及相互的交互。IPDD 模型面向信息服务架构进行描述，因此，界面原型驱动开发的核心环节是依据界面原型探索信息服务的底层实现。因为需求阶段已将信息特征映射为服务描述接口，所以依据服务接口描述完善该服务的具体实现即可，并以此为基础进行多次迭代开发，直到交付阶段性成果。

基于界面原型驱动开发模型对相关界面元素隐含的功能条目和约束代码以 Web service 组件模式来实现，是面向信息服务架构的界面原型驱动开发的核心。如在界面原型 IP-1 中，从界面视图的输入栏输入某一书籍名称，点击“搜索”按钮，该界面图元即可驱动功能需求条目 FR-1 的设计和实现，从而完成界面原型驱动开发过程。然后，针对特定功能需求的服务代码进行迭代，期望每 2 个星期完成一次迭代，并快速交付客户可见的中间结果，向客户呈现期望的软件原型。最后，逐步实现整个界面图元驱动的开发。

5 Cyber 书社实例分析

本节重点结合上述 Cyber 网络书社的例子，以面向信息服务的 IPDD 模型为指导，说明该系统典型功能需求的抽取、界面原型的设计及基于界面原型驱动设计和实现过程，并对该模型指导开发的性能、效益进行综合评价。

5.1 Cyber网络书社架构

在面向信息服务的 IPDD 模型指导下,为了提高系统的灵活性和可扩展性,Cyber 网络书社在平台组件的构成中使用面向信息服务架构,重点应用 Web service 相关技术构建和实现,通过前期调研和分析,该书社逻辑上由客户服务网站和商品配送中心 2 个主要部分组成。

借助前期的需求调研和分析,应用 IPDD 模型指导,初步确定 Cyber 网络书社的功能需求内容。根据需求分层模型理论,并应用功能需求获取过程规约,分析得出该项目的 2 个核心功能需求为信息浏览和查询及购物车和订单确认。

在该书社平台功能需求模块中,功能条目“书籍查询及购买”对应为 4.2 节中的功能需求 FR-1。限于篇幅,其他的需求内容和具体的需求描述不再赘述。

5.2 Cyber书社用例分析及服务实现

Cyber 网络书社系统架构已经明确,需求分析和获取基本完成,下面进一步对 Cyber 书社的顶层用例进行分析。在该项目中,用户类主要为网站客户和管理员,基于用例模型,分析得到 Cyber 网络书社顶层用例,如图 5 所示。

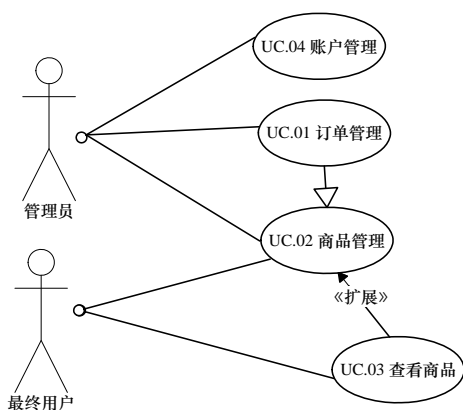


图 5 Cyber 书社顶层用例

进一步对用例图中编号为 UC.03 的用例进行分析,则可获取基于 UC.03 的全部功能,其中也包含前述分析得到的功能需求 FR-1。

在面向信息服务的 IPDD 模型指导下,应用 4.1 节中对话图的相关理论基础,基于功能需求 FR-1 进行对话图模型分析,即得到对话图例 DB-1,依据面向信息服务的 IPDD 模型中界面原型建模的过程和方法,基于 DB-1 设计出界面原型图例 IP-1 和 IP-2,基于界面原型图例即可进一步驱动特定图元下的功能实现,即相应服务接口的实现。

界面原型驱动的本质是在忠实于功能需求的基础上,依据面向信息服务的 IPDD 模型指导完成全面的分析、设计和编码工作,并基于功能映射的服务接口执行实现和迭代过程。在 IPDD 模型指导下,针对功能需求 FR-1 中的“请求指定书籍”功能,基于其服务接口描述,设计出其界面图元为 IP-1 中的“搜索”控件,另一个则是 IP-2 中的“购买”控件。前者将驱动一个搜索指定书籍的服务接口的设计和实现,后者则驱动一个“下订单”的服务接口实现。限于篇幅,将查询指定书籍过程中定义的相关变量及执行存储过程等相关细节略去,则实现该功能的简易服务代码如下:

```
SearchBook
[WebMethod(Description="查询指定的书名")]
public XmlDocument GetSiteAData(string BookName)
{
```

```
    XmlDataDocument xd=new mlDataDocument();
    DataSet ds=new DataSet();
    CStoreProc cp=new CStoreProc("SearchBook"); cp.AddPar
In("@keywords",SqlDbType. VarChar, 30, BookName);
    cp. AddParOut("@res", SqlDbType. Int);
    if(cp.SelectProc()) //如果执行成功, 存储过程
    { cp. myData.EnforceConstraints=false; //不严格格式检查
    if((int)cp.GetReturnValue("@res")==-1)
    { string xml="";
    xd.LoadXml(xml);
    return xd; }
    xd=new XmlDocument(cp. myData);
    XmlNode root1=xd. DocumentElement;
    XmlNodeList roots=root1. SelectNodes("list");
    foreach(XmlNode roota in roots) //为元素加上站点标记
    { XmlElement Link=xd. CreateElement("SiteName");
    Link. InnerText=ConfigurationSettings. AppSettings
["SiteName"]. ToString();
    roota. AppendChild(Link); }
    return xd; }
    else return null; }
```

然后,基于该服务代码进行逐步迭代,交付可见的中间成果,直至完成一轮典型的 IPDD 驱动开发过程,最终完成其他的所有功能需求条目,并交付高质量的系统。

面向信息服务的 IPDD 模型强调灵活的面向服务架构,快速响应需求变化和进行原型迭代过程,从而赢得较为宽松的项目进度要求。特别是在界面原型设计过程中,可以直接、快速地向客户交流,获得客户的反馈,并迅速修正错误,最终赢得客户较高的满意度。

应用新模型也存在一些明显的不足,如需求条目可跟踪性较差、文档不完善及质量较低。

6 结束语

面向信息服务的 IPDD 模型以需求分析为基础,以界面原型建模和驱动开发为核心,快速进行原型迭代,完成一轮从需求到设计的完整开发过程。实例分析表明,该模型具备一定的理论基础并符合工程原则,支持快速响应业务需求的变更,提高了以数据为中心的信息管理系统分析设计质量和效率。

参考文献

- [1] 李建华, 陈松乔, 马 华. 面向服务架构参考模型及应用研究[J]. 计算机工程, 2006, 32(20): 100-102.
- [2] Newcomer E, Lomow G G. Understanding SOA with Web Services[M]. 徐 涵, 译. 北京: 电子工业出版社, 2006.
- [3] 荣辉桂, 徐小林, 陈 浩. 基于概念测试用例的快速需求测试过程模型[J]. 计算机系统应用, 2007, 28(8): 76-80.
- [4] Rong Huigui, Zhou Ning, Jin Min, et al. Research on Service-oriented Framework of Interface Prototype Driven Development[C]//Proceedings of IEEE International Zurich Seminar on Computer Science and Software Engineering. Zurich, Switzerland: IEEE Press, 2008: 552-557.
- [5] Weigers K E. 软件需求[M]. 2nd ed. 刘伟琴, 刘洪涛, 译. 北京: 清华大学出版社, 2006.
- [6] 于 卫, 楼顺天, 杨万海. 图形用户界面的形式化描述方法[J]. 西安电子科技大学学报, 1998, 25(4): 426-429.

编辑 张正兴