

# 基于服务总线的舰载指控系统集成方法研究

(中国船舶重工集团公司第七〇九研究所、武汉、430074)

**摘要:** 针对新一代舰艇指控系统对软件集成能力和水平的要求, 本文提出了基于服务总线的舰载指控系统集成方法, 分析了服务总线的体系构架, 实时服务总线的模型, 以及面向服务的舰艇指挥控制系统服务化体系结构和面向服务集成的实施方法。

**关键词:** 舰艇指控系统 实时服务总线 面向服务的构架 业务流程

## 0 引言

舰载指控系统从诞生以来先后经历了独立式、集中式、分开式和分布式四个演化阶段, 并随着信息技术与作战方式的不断发展, 正逐步发展为支持“网络中心战”的全分布式的综合指控系统。与传统舰载指控系统相比, 新型舰载指控系统不但要具有作为独立平台单独执行作战任务能力, 还要能够进行编队、护航等多兵种协同作战, 这就要求指控系统具备动态可配置、可重用、可功能互操作等特性, 通过动态的资源管理和业务流程柔性重组来满足不同作战模式及作战任务需求, 同时对舰载指控系统的集成能力和水平提出了更高的要求。本文从探讨了基于服务总线的舰载指控系统的服务化集成方法,

## 1 服务总线

### 1.1 服务总线体系构架概述

嵌入式服务总线是舰载作战指挥系统面向服务构架的基础设施和实现平台, 充当连接和集成各种实时/非实时应用系统消息总线, 实施了必要抽象层将定义服务的消息转化成可供业务构件处理的数据。实质上是服务间的连接框架, 核心功能包括了消息转换、消息机制、基于内容的路由和服务容器等四部分。

#### 1)消息机制

消息机制提供管理计算资源和网络通信的机制, 屏蔽分布环境复杂性和异构性, 为应用程序提供透明的通信服务。ESB 的消息机制采用通信通道(channel)抽象服务之间的消息通信, 服务之间建立通道联系。ESB 可支持两种通信模式: 发布/订阅和点到点的请求/回复的消息模式。

发布/订阅是异步消息传递模式, 发布者发布的消息可传递给多个订阅者。在发布/订阅模式中, 首先订阅者将订阅信息发布到 ESB, 发布者在发布消息后, ESB 将消息转发给相关订阅者。发布/订阅方式一般通过主题树(Topic Tree)实现。主题树以信息发布者为根接点, 父、子结点之间表示了一种发布/订阅关系。每个父结点将消息发布给其子结点; 每个子结点可以选择接受或拒绝来自父结点的所有消息。

请求/回复模式是服务提出请求, 其它服务响应回复的模式, 每个消息仅传递给一个消费者。它可以是同步也可以是异步的。同步方式中, 请求方等待回复以进行后续操作; 异步方式中, 请求方无需等待回复消息。请求/回复模式可以有单向/双向两种消息通道。单向通道只传递请求或回复消息; 双向通道中一个通道可同时传递请求和回复消息 请求/回复一般通过队列实现。每个服务都可建立其请求和回复队列。

#### 2)消息转换

消息是自包含、自治的实体。一个消息通常由消息头、消息属性以及消息体三个部分构成。自包含消息定义是解耦合(避免两个服务之间直接通信)、实现异步消息机制的关键。通常 ESB 选择已有的企业数据模型或工业标准消息作为内部通用的基于 XML 的标准通信协议和消息格式, 例如目前电子商务中的 xCBL 格式、ebXML 等

连接在总线上的服务种类很多, 可能采用不同的消息协议, 其对于信息的需求也不同, 例如两个通讯的服务 A 采用时间格式是“年年年年一月月一日日”, 而 B 采用“月月一日日一年年年年”。因此, 需要对消息进行转换。消息转换包括消息通信协议的桥接和消息内容转换。协议桥接实现不能直接通讯的协议之间消息的传递, 内容转换支持对不同消息内容的转换。IBM 还提出, ESB 还可以补充、完善消息内容, 例如在消息中添加消息的来源等信息。

消息桥接有三种实现方式: a) 通过一个共享文件缓冲区暂存交互的消息, 消息交互方可采用不同的协议, 往缓冲区中存、取消息; b)通过第三方标准协议, 如 JMS、JBI、SOAP 等; c)ESB 直接提供对不同消息协议的处理能力, 如将 HrrP 或 SOAP 请求映射到 JMS 消息。此外 ESB 可采用例如 XSLT(eXtensible

Stylesheet Language Transformation)技术,通过 Schema 映射的方式,实现对不同的 XML 消息格式的转换。

### 3)消息路由

消息路由是分析服务传递的步骤,建立传递线路和规则,并逐步传递消息的过程。ESB 可根据消息内容将其由提供者传递到接受者。消息路由主要包括路由线路和路由规则两个部分。

路由线路描述了服务将要发送的地址和路由规则,包含在消息的元数据中。它分为不带分支判定的简单消息线路和包括分支、判定、条件等执行过程的复杂消息线路。简单消息线路上的各个服务结点顺序执行;复杂处理过程在复杂的路由中,需要支持流程分离、流程聚以及复杂分支判定等多种处理机制,通常用服务 workflow 语言如 BPEL4WS 来描述。

ESB 路由规则多采用分布式的管理方式,即规则分散在各个服务节点,由该节点的服务容器管理。路由规则定义了消息传递和路由线路的选择策略,例如需要满足的前提条件、路由时间延迟要求、可允许的失败连接的次数等。在复杂路由线路选择中,路由规则为其提供决策依据。

### 4)服务容器

服务容器是将各种类型的软件组件或应用,封装成可支持标准通讯协议(如 JMS、JBI、JCA、Soap 等)的服务,并抽象成一个端点(endpoint),连接到总线上的组件。服务容器既可以封装用户应用软件,也可以封装 ESB 的基础服务。

为了实现分布式处理,服务容器需要支持服务的注册、发现和选择。有两种方式实现服务发现:一种方式是服务容器本身不管理注册信息,通过查询总线全局注册表,获取服务信息;另一种方式是服务容器维护一个本地注册表,通过本地查询即可获得服务信息。

通过服务容器,可以实现对软件的局部管理和全局管理相结合的方式。局部管理是服务容器对其所封装的软件的管理,包括生命周期管理、连接管理等。ESB 可以通过标准协议如 JMX(Java Message eXtension)框架、SNMP(Simple Network Management Protocol)等集成其它管理工具(如 IBM Tivoli, HP Openview 等),实现对服务容器的全局管理,如远程配置管理、路由规则管理等。通过服务容器管理,可实现容器选择性功能布置。

服务容器屏蔽了软件的异构性,使得总线的基础服务对每个特定的服务软件透明。对应用软件而言,容器是总线的代理,服务容器是基于 ESB 的系统可扩展性的重要保证。

## 1.2 现有服务总线模型及其存在的问题

Java 业务集成(Java business integration, JBI)是 JCP(Java community process)制定的 Java 平台的 ESB 标准化技术规范。JBI 中定义了两种组件:绑定组件(binding components, BC)和服务引擎(service engine, SE)。绑定组件用于根据特定协议和传输器发送和接收消息,如 HT\_TP / Web Services、JMS、FTP 等。服务引擎实现业务逻辑,编排整合各种服务。绑定组件把消息在特定协议的格式和规格化格式之间进行转换,使 JBI 系统只需处理规格化消息,从而把 JBI 系统与特定协议分离。服务引擎和绑定组件通过传输通道(delivery channel, DC)与标准化消息路由(normalized message router, NMR)进行通信。

JBI 给出了 ESB 系统的接口和交互标准,但并没有给出具体的实现方法,例如如何应付高并发的业务请求。一些基于 JBI 的 ESB 系统采用部署集群的方式应对高并发的业务请求,但由于接入总线的外部 Web Service 服务本身的处理能有限,即便配置再多集群,对该服务的调用仍然是多种业务流对有限资源的竞争使用。还有一些系统基于 SEDA(staged event-driven architecture)架构[8]。业务请求被放置在缓冲队列中,采用先到先服务和尽力工作的服务策略。这两种实现方式都没有考虑到不同类型的业务对时效性的要求。当系统重载,总线发生业务拥塞时,所有的服务都处于等待状态,系统服务质量迅速下降。

ESB 作为一个集中式数据和服务交换平台,需要处理大量的实时业务。实时业务对业务的执行时间有明确的时间约束。ESB 应具备实时调度能力,尽可能地保证每个任务满足他们的时间约束。一方面对外部请求做出及时响应,另一方面缩短系统平均响应时间,提高系统资源利用率。

## 1.3 实时服务总线模型

表 1 三种不同级别业务类型

业务类型	业务特点	实时性	优先级
第一类	有较小时延和抖动,具有固定传输速率	实时	高
第二类	具有可变速率,相对第一类业务对时延和抖动要求较低,需要最低带宽保证	非实时	中
第三类	对时延和抖动基本无要求,系统重载时可以暂停服务	非实时	低

ESB 作为一个集中式数据和服务交换平台，需要处理大量的实时业务。实时业务对业务的执行时间有明确的时间约束。ESB 应具备实时调度能力，尽可能地保证每个任务满足他们的时间约束。一方面对外部请求做出及时响应，另一方面缩短系统平均响应时间，提高系统资源利用率。

ESB 作为一个集中式数据和服务交换平台，需要处理大量的实时业务。实时业务对业务的执行时间有明确的时间约束。ESB 应具备实时调度能力，尽可能地保证每个任务满足他们的时间约束。一方面对外部请求做出及时响应，另一方面缩短系统平均响应时间，提高系统资源利用率。

为了使 ESB 系统能够处理实时业务，对 JBI 标准进行改进，采用优先级驱动策略，按照业务优先级的高低动态分配各类业务流的带宽。

首先为基于 JBI 的绑定组件增加两个语义。其一是优先级 Priority，取值为 high, middle 或 low。其二是业务时间约束 timeoutMS，单位为毫秒。当业务排队等待时间超过该约束时，业务将返回超时错误。

```
<http: soap-provider
service=" hospital: DiagnoseServiceSOAPBinding"
wsdl=" classpath: diagnoseService. wsdl"
priority=" high"
timeoutMS=" 30000" />
```

接着扩展 JBI 的标准化消息路由组件，增加下列 3 个组件，如图 1 所示。

(1)流量整形漏桶组件。该组件位于服务提供者组件和消息路由组件之间，采用漏桶算法(leaky bucket, LB)对业务请求进行流量过滤和整形。如图 1 所示，业务请求先被一个输入缓冲队列中，漏桶流控组件根据系统分配给该业务的流量带宽将业务消息放入输出缓冲队列中供消息路由组件分发。如果业务请求因为拥塞排队超过有效时间，漏桶流控组件直接返回超时错误消息给绑定组件。

(2)流量优先级调度组件。该组件位于服务消费者组件或服务引擎组件和消息路由组件之间。如图 1 所示，该组件为每一个业务流建立一个传输队列，消息路由将不同的业务流消息放入相应传输队列中。调度组件使用加权公平队列算法(weighted fair queuing, WFQ)，以分配给该业务的带宽作为权重调度各业务流，将消息路由至其它服务消费者组件或服务引擎组件。WFQ 和 LB 算法共同作用保证总线上各业务流量的稳定。

(3)消息路由组件。该组件定时与流量整形漏桶组件通信，实时获取并记录各业务输入流量、业务超时和丢弃情况。当服务提供者组件发生流量拥塞时，该组件能根据总线当前流量情况动态调整分配各业务流带宽。

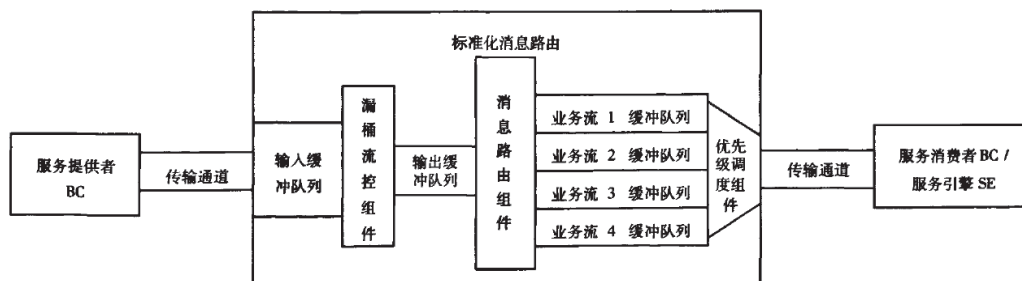


图 1 实时服务总线消息处理

在实时 ESB 模型中，一次完整的业务流程如图 2 所示。假设客户端通过 Web Service Client 以同步方式调用 HTTP / SOAP BC。该调用在进入总线后变成异步模式的消息。BC 将 SOAP 协议的调用请求转换成标准化消息，通过传输通道放入 BC 对应的 LB 组件的输入缓冲队列中。LB 组件根据分配的流量按指定速率将消息发送给 NMR。如果业务请求因为拥塞排队超过有效时间，LB 直接返回错误消息给 BC。NMR 将消息路由至目的 BC 对应的 WFQ 组件，放入相应业务流的缓冲队列中。WFQ 组件根据优先级调度算法将请求发送给消费者 BC 组件。消费者 BC 组件调用外部系统相应的接口后将消息直接返回给标准化消息路由，标准化消息路由将消息返回给业务发起的 BC。BC 将标准化消息转换成 SOAP 协议后返回给阻塞的客户端。

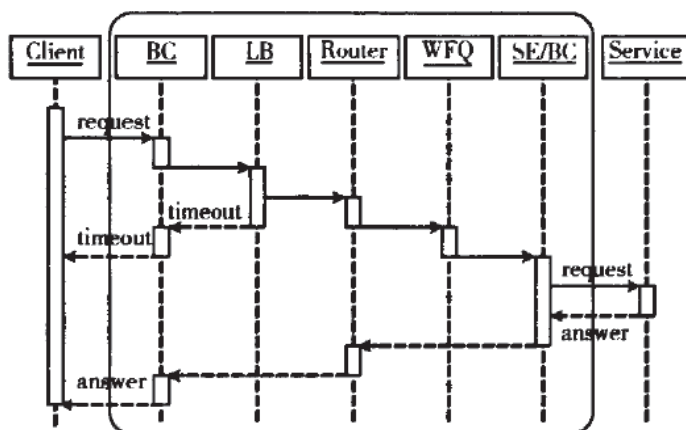


图2 消息序列

## 2. 面向服务的舰艇指挥控制系统服务化体系结构

依据舰艇作战指挥系统的功能需求及其体系结构分析, 本文构建了“以服务为核心”的多层次模型; 并以分层表达方式将舰艇指控系统体系结构划分为: 平台系统层、基础构件层、服务层、业务服务层、表示层共5层。该结构基于 Web Service 体系结构, 充分体现了面向服务的“松耦合”、“即插即用”、“多重复用”、“灵活重组”等特点, 是开放式系统结构体系; 能有效提高潜艇指控系统的技术适应能力、扩展能力、升级适装、互操作能力。具体见图3。

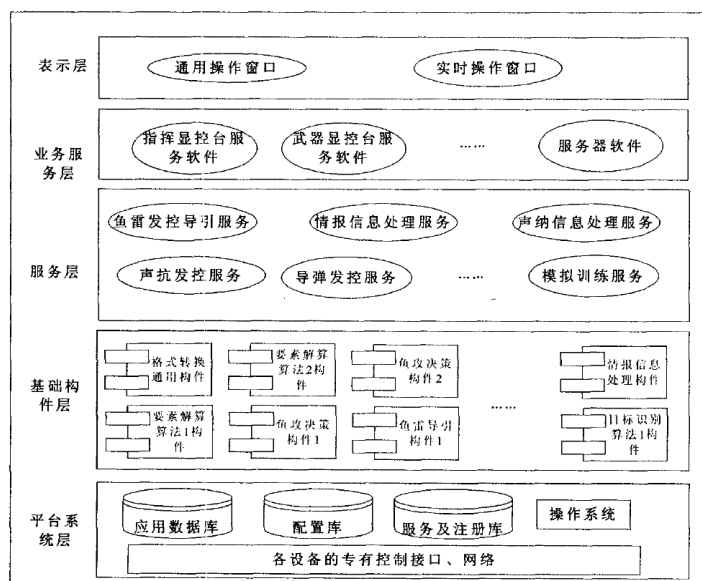


图3 以服务为核心的多层次舰艇指控系统体系结构图

### 1) 平台系统层

主要是指控系统所属分系统的设备及其软硬平台, 如军用加固显控台、加固机柜、数据库、操作系统、板卡驱动程序和计算机网络等软硬环境。其中数据库包括战术数据库、专家知识库、配置库、服务及注册库; 操作系统包括 Windows、Linux、Vxworks 及国产实时操作系统。

### 2) 基础构件层

基础构件层包括专有构件与通用构件两种。专有构件指为专有设备服务, 只能部署到专有控制设备绑定后才能使用, 如战术数据库构件、专家知识库构件必须部署在服务器上; 通用构件指不受平台、软硬环境限制, 可以在通用平台上部署使用, 完成特定功能的构件。这类接 13 主要包括各种信息处理构件、目标要素解算算法构件、鱼雷攻击决策构件、鱼雷导引算法构件等。

### 3) 服务层

服务层是指潜艇指控系统服务集, 包括原子服务与组合服务。通过 Web Service 封装基础构件及专有构件的服务为原子服务, 通过将原子服务按照 BPEI, 4WS 组合成完成声抗发控、导弹武器发控等一定功能的服务称为组合服务。该层可根据平台装备配置及具体作战任务, 配置出相应的功能服务。

#### 4) 业务服务层

这里的业务主要针对作战业务部门使用者，具体对应于作战站位。业务服务层把组合服务层的各种服务，根据作战任务及作战流程，按照站位进行打包、配置和封装，以供最终作战用户使用。该层对应于指控显控台、武器显控台、情报显控台等设备上软件功能。

5) 表示层 按照“软硬件分离、数据与应用分离、显示与处理分离”的原则，表示层提供标准的人机操控界面，将需要展现的数据展现给操作使用人员。根据使用需求不同划分为通用操控界面与实时操控界面。通用操控界面为所有非实时需求的作战设备提供人机操控与数据显示，通过解析与控制业务层传送 XML 数据与数据处理分离；对实时数据显示有严格要求的实时操控界面则采用 X Windows 实现。

BPEL4WS(Business Process Execution Language for Web Services, BPEL4WS)定义了一种基于 Web Service 的业务流程行为的表示方法，是一种使用 Web Service 定义和执行业务流程的语言。本文用来解决多层次体系结构模型中服务层、业务层两层中原子服务及组合服务的组合、编排、配置以及部署。具体过程见图 4。

依据上工作流程，潜艇指控系统可根据作战任务需求与系统配置不同，利用 BPEL4WS 调整作战流程，重新进行软件上的编排、配置及部署。该技术与传统的指控系统相比，各服务软件不需要修改代码、不需要重新编译、不用更改硬件，极大减少了开发进度、减少系统开发的时间和成本，且系统维护简便。

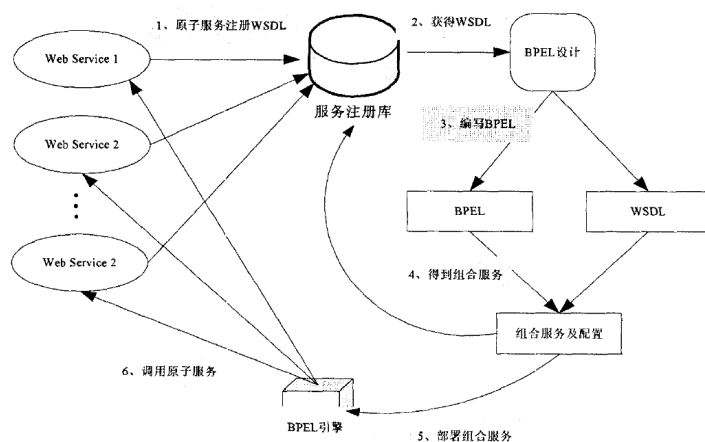


图 4 基于语义的 BPEL4WS 服务组合工作流程图

### 3. 基于服务总线的舰艇指控系统集成实施方法

依据构建的面向服务的舰艇指控系统软件体系结构，指控系统的软件开发就要打破原有的开发模式，需要有新的开发方法及开发过程，具体实施方案见图 5。

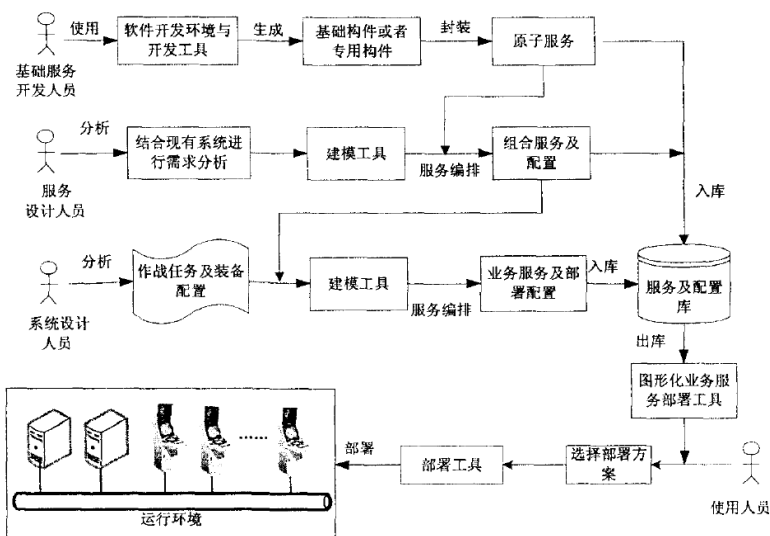


图 5 舰艇指控系统服务化软件开发及部署图

实施方案将舰艇指控系统软件开发人员分为：基础服务开发人员、服务设计人员、系统设计人员三类。三类人员具体工作方法如下：



为继承现有成果,基础服务开发人员可以将现有的软件或构件按照 Web Service 标准封装原子服务,或者按照 Web Service 标准重新开发新的原子服务,开发结束后入库提供给上一级使用;服务设计人员采用自顶向下的分析方法得到各功能所需的原子服务,按照 BPEL4WS 服务编排方法得到组合服务及相关的配置文件,统一入库供系统使用;系统设计人员根据作战任务及装备可能的配置情况,利用建模工具得到各种方案的业务服务及部署配置,以图形化的部署方式,供用户使用。

#### 4. 结束语

本文针对舰艇指控系统所面临的易于实现与集成的实际作战需求,构建了“以服务为核心”的多层次体系结构模型,并在此基础上提出了工程实施方法及技术解决方案。研究成果对实现具备即插即用、柔性重组、良好的互操作的服务化舰艇指控系统,具有重要工程应用价值。

#### 参 考 文 献

- [1] 杨芙清,梅宏,李克勤. 软件复用与软件构件技术[J]. 电子学报, 1999, 27(2): 68—75. (Yang Fuqing, Mei Hong, Li Keqin. Software reuse and software component technology[J]. ACTA electronica sinica, 1999, 27(2): 68—75)
- [2] 郭煦. 基于构件的软件生产模型[J]. 计算机应用与软件, 2010, 27(7): 156-158. (Guo Xu. Component-based software development model[J]. Computer applications and software, 2010, 27(7): 156-158.)
- [3] 杨晓梅. 如何开展软件的质量管理[J]. 现代雷达, 2005, 27(11): 81-84. (Yang Xiaomei. How to carry out quality management software[J]. Modern radar, 2005, 27(11): 81-84.)
- [4] 傅程, 洪慧. 军用软件构件可测试性度量方法研究[J]. 装甲兵工程学院学报, 2007, 21(2): 82-86. (Fu Cheng, Hong Hui. Research on measurement for military component-based software testability[J]. Journal of academy of armored force engineering, 2007, 21(2): 82-86.)
- [5] 毛国蓓, 李雪静, 葛孝堃等. 基于软件构件质量模型的度量及应用[J]. 计算机应用与软件, 2005, 22(5):1-4. (Mao Guobei, LiXuejing, Ge Xiaokun. The metrics and application of software-based component quality model[J]. Computer applications and software, 2005, 22(5):1-4.)
- [6] ALAN W B. 大规模基于构件的软件开发[M]. 美国普林提斯.霍尔出版社,2000: 59 (ALAN W B. Large-scale, component-based development[M]. USA: Prentice Hall PTR, 2000: 59)
- [7] 张世琨, 张文娟, 常欣等. 基于软件体系结构的可复用构件制作和组装[J]. 软件学报, 2001, 12(9): 1351—1359. (Zhang Shikun, Zhang Wenjuan, Chang Xin. Building and assembling reusable components based on software architecture[J]. Journal of software, 2001, 12(9): 1351—1359.)
- [8] 颜世刚, 张振中. 基于 SaaS 的军用软件开发模式研究[J]. 微处理机, 2012(1): 49-55. (Yan Shigang, Zhang Zhenzhong. Research on the military software development's node based on SaaS[J]. Microprocessors, 2012(1): 49-55.)
- [9] 吕明琪, 薛锦云, 胡启敏. 基于软件体系结构的可复用构件模型[J]. 计算机应用研究, 2008, 25(1): 120. (Lv Mingqi, Xue Jinyun, Hu Qimin. Reusable component model based on software architecture[J]. Application research of computers, 2008, 25(1): 120.)
- [10] 刘大昕, 王卓, 冯小宁. 支持集成的软件构件库设计与实现[J]. 计算机工程, 2005(11): 225—227. (Liu Daxin, Wang Zhuo, Feng Xiaoning. Design and implementation of software component Repertory supporting integration[J]. Computer engineering, 2005(11): 225—227.)