分　类　号：TP311.52　　单位代码：10183
研究生学号：200853H002　　密　　级：公 开

吉 林 大 学

硕士学位论文

基于开源 ESB 的校级统一信息系统实现

Implementation of the University Level Unified Information System Based on Open Source ESB

作者姓名：蔡珉官
专　　业：软件工程
研究方向：中间件技术
指导教师：刘磊　教授
培养单位：计算机科学与技术学院

2011 年 4 月

基于开源ESB的校级统一信息系统实现

Implementation of the University Level Unified Information System Based on Open Source ESB

作者姓名：蔡珉官

专业名称：软件工程

指导教师：刘磊　教授

学位类别：软件工程硕士

答辩日期：2011 年 5 月 21 日

吉林大学硕士学位论文原创性声明

本人郑重声明：所呈交的硕士学位论文，是本人在指导教师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：蔡琨宜

日期：2011 年 5 月 21 日

摘　要

基于开源ESB的校级统一信息系统实现

随着高校信息化步伐的不断加快，高校信息化从应用系统的独立建设阶段步入到各种应用系统的集成阶段，从各部门分散建设阶段提升为跨部门应用集成阶段。如何建设各种校级统一的应用系统是目前各个高校面临的共同挑战。而作为下一代数字校园建设的重要组成部分，建立灵活的统一应用系统架构被成为关键所在。近几年很多高校已经把高校信息系统集成工作放在重要的位置来建设，但其进度非常缓慢，效果不太明显。开源集成方案的不断涌现为很多企业，尤其是为一些资源不充足的地方高校，增添了关于系统集成的可能性。

SOA（面向服务的架构）是一种帮助系统在增长的同时保持可扩展性和灵活性的方法，也是目前比较成熟的应用系统集成架构。SOA主要元素之一的ESB（企业服务总线）是个专门的基础设施，它能够把各种服务以非常灵活的方式结合起来。市场上早已出现了各种成熟的 ESB 产品，其中最受人关注的还是开源 ESB 产品。因开源软件本身的灵活性等特点，越来越多的组织和机构采用开源 ESB 产品，在ESB的开源社区中很容易找到各种适合自己的解决方案。

本文是作者参与延边大学建设校级统一信息系统项目时研究和学习的成果。本文首先介绍了与本课题相关的各种理论基础。该部分首先介绍了SOA概念和实现关键技术。接下来的部分重点介绍了ESB技术，主要内容包括ESB的概念、特点以及优势。在ESB的开源技术方面重点介绍了Mule ESB产品架构。在本文的接下来部分介绍了信息系统集成中的设计模式。该部分主要介绍了基于ESB的信息系统集成中常用的几种设计模式。第四部分是本文的重点。在该部分中首先分析了高校在系统集成方面面临的挑战，并根据问题做出校级统一信息系统的需求分析。在需求分析过程中把SOA思想融入到校级统一信息系统实现过程，把数据同步思想作为实现 ESB 的关键过程。第五部分是根据需求分析进行了基于 ESB 的系统集成设计工作。在基于ESB的设计过程中采用了各种集成设计模式。第六部分是基于Mule的校级统一信息系统的具体实现过程，介绍了实现过程中的部分关键性代码和解释。本文最后部分将总结前期的一些研究工作，为今后的集成

工作提出了方向。校级统一信息系统的建设是一个长期的集成过程，本文的主要研究范围锁定在了基于 ESB 的基础设施搭建方面，而且在整个校级统一信息系统建设过程中占非常小的一部分。真正实现校级统一信息系统还需要更多方面的集成技术研究。

关键词：

面向服务架构，企业服务总线，Mule，设计模式，校级统一信息系统

Abstract

Implementation of the University Level Unified Information System Based on Open Source ESB

Accelerating the pace of Informationization in colleges and universities, independent entered the construction phase of college application information from the system to various application systems integration stage, from the scattered construction phase promotion across Department application integration stage. How to build a variety of school-level unified system is the individual colleges and universities face common challenges. And as an important part of the next generation of digital campus construction, establish a flexible uniform application of the system architecture is becoming crucial. In recent years, many universities have integrated information system in colleges and universities in an important position to build, but their progress very slowly, the effect is less obvious. Emergence of the open source integration solution for many businesses, especially for a number of resources are not adequate local University, was added on system integration possibilities.

SOA (Service-oriented architecture) is a help system to grow while maintaining scalability and flexibility of methods, is also more mature application system integration framework. One of the SOA main elements, ESB (Enterprise Service Bus) is a specialized infrastructures, it can combine various services in a very flexible way. Markets have already become a variety of mature ESB products, most of them to be concerned is the open source ESB products. Due to the flexibility of open source software, and more and more organizations to adopt open source ESB products, ESB open source community can be found in a variety of suitable solutions.

This article is that the authors involved in the construction of Yanbian University's University level unified information system project studying and learning outcomes. This article first describes the theoretical basis of the related. This section

first describes the SOA concepts and key technologies. The next section focuses on the ESB technology, main content including ESB concepts, features and advantages. About the ESB's open source technology, highlights Mule ESB's product architecture. In the next section of this article describes the design of information system integration mode. This section mainly introduces the ESB in information system integration of commonly used design patterns. Part IV is the most important part of this article. In that part of the first, analysis the universities challenges faced by in the system integration and make a demand of University level unified information system based on problem analysis. During the needs analysis process to SOA thoughts into school level information system implementation process, thinking as the implementation of data synchronization ESB key processes. Part v is based on requirements analysis based on ESB system integration design. There is a variety of integration design patterns usage based on ESB in the design process. Part VI is based on the Mule school level implementation process of a unified information system, introduced in the process of achieving some of the key code and explanations. The last section of this article will summarize the early works, make the direction for future integration work. School-level building is a long-term process of integration of a unified information system, locked in a major research areas of this article is based on the ESB infrastructure to build, but also throughout school unified information system construction in a very small part of the process. Real school level also need more research on integration technology of unified information system.

Keywords:

SOA, ESB, Mule, design pattern, University level unified information system

目　录

第一章 绪　　论

1.1 引　言

1.1.1 研究背景

由于经济发展等原因，企业信息化一直领先于教育信息化。尤其是发达国家的制造业等领域，其信息化建设尤为先进。企业信息化经历了三个阶段的发展：建设分散独立的信息系统阶段、建设全局性统一信息系统阶段、全面提升统一信息系统阶段[1]。我国高校信息化经历了 20 多年的长足发展。上世纪 80 年代末到 90 年代初，主要是单机版的应用为主，解决重点领域的电算化。上世纪 90 年代中期到 2000 年代初是我国高校网络的大发展时期，各种以部门为主的网络版信息系统大量的涌现，推进了高校各项管理工作的信息化。但信息系统之间的关联非常有限，有的系统甚至与其他系统出现数据冲突，无法实现真正意义上的信息共享。随着我国教育信息化建设的不断深入，建立全局性信息系统的需求不断升级，一些信息化程度较高的高校已经建立了统一系统集成平台，有的高校甚至已经进行业务流程整合相关的全面系统提升工作。我国的教育信息化已经开始跨入到建设全局性信息系统阶段[2]。

全局性信息系统在高校信息化当中称为校级统一信息系统，也就是学校统一组织建设的全局性信息系统，能支撑学校跨部门跨领域的各项业务流程。校级统一信息系统建设是一个漫长的过程，也是长期需要努力的工作，一般细分三个阶段：应用集成阶段、信息集成阶段和业务集成阶段[3]。其中第一阶段是实现第二、第三阶段的准备阶段也是目前大部分高校需要解决的重要工作，主要包括基础设施整合、数据整合、用户整合以及应用整合[3]。如何建立一种灵活的、安全可靠的应用集成架构是后续所有校级统一信息系统建设的关键。

企业应用集成，即 Enterprise Application Integration（EAI），是指在企业范围内，将多个应用系统的过程、软件、标准和硬件集成起来，使其成为无缝运作的整体[4]。传统企业应用集成解决方案，其投资力度大、所用协议不标准、硬件平台要求高等特点，目前在高校信息系统集成领域发展非常缓慢。尤其

是在灵活性等方面受到重大的挑战，难于满足学校业务变化产生的各种需求变化。SOA（Service-oriented architecture，面向服务架构）很好的解决了上述EAI 面临的问题，目前应用在很多系统集成领域，在业内得到了充分的认可。

SOA 比较接近于企业内部的软件资源，在内部网络中可发现的软件资源在SOA 中定义为服务。每种服务对应着特定的业务流程，同时这些服务之间的工作是互不相关的。这些服务是互相独立的，不依赖于其它服务的上下文或者状态。早期的 SOA 是基于 CORBA 的 COM 或者 ORB 来实现的，但目前随着 Web Service 技术的发展常用的实现技术转变为 WSDL,UDDI 以及 SOAP 等等。除了 Web Service 外，其实还有很多 SOA 的实现技术，尽管 Web Service 有很多不成熟的部分，但到目前为止它始终是实现 SOA 的最简单、最有效的途径。

随着开源产品的不断涌现，在高校未来的 IT 架构规划中需要更多的考虑开源产品。SOA 的开源平台近几年发展的非常活跃，经过多年的发展在 SOA 的各实施领域都出现了丰富的开源解决方案。比如基础设施方面有 Mule、Service Mix 等等，在 Web 服务的搭建方面出现了 SCA 等优秀的开源框架，在业务流程方面出现了成熟的 jBPM 开源框架。SOA 开源平台的搭建不仅节省了系统集成的成本，而且在具体的实施当中充分发挥了灵活性的优势。同时随着开源社区的不断活跃和壮大，在实施过程中所碰到的问题被解决的可能性不断加大。

ESB 作为 SOA 实施过程的一个重要工具，主要承担互操作的功能，它提供了消息传输、转换、路由、安全等等一系列功能，能让所暴露出来的 web 服务灵活的使用在业务流程整合当中。可见 ESB 在 SOA 的实施过程所占的比重和重要性非常大，当然开源产品也不放过这么重要的领域。在 ESB 领域有很多优秀的开源产品，包括 Mule、Service Mix、Open ESB 等等，而且它们的支持文档也日益健壮，不断完善。

1.1.2　研究目的和意义

随着高校信息化步伐的加快高校各部门的信息系统数量也逐渐增多。这些各部门的信息系统很难无缝的结合在一起，也就是说经常出现不兼容、升级性和灵活性的问题。而目前SOA在系统集成领域里发展的非常迅速，已经得到了很多业界的认可，它将应用系统抽象成一个个粗粒度的服务，标准化服务接口，降低服

务之间的耦合度。传统软件的许可证费用不仅增加了成本也限制了服务集成的灵活性。而作为经费不足的地方高校，转向开源软件技术有助于缓解这些问题和加快服务集成实施。开源应用与SOA越来越成为完美的搭档，很多SOA实现方案中对开源技术的使用受到了广泛的关注。而作为SOA平台的基础设施部分ESB开源软件目前也迅速崛起，已经在SOA平台的搭建中起到重要的角色。事实上，今天已经有很多ESB开源软件及开源服务可供使用了。

1.2 国内外研究现状

高校的系统集成问题多年来一直是业界关注的问题，目前市场上很多商业的解决方案，比如在数据方面有数据交换平台和数据同步等等，在业务流程整合方面有全局业务搭建等等。一些提供大型的 IT 解决方案的公司，例如 Oracle、IBM，已经能提供成熟的一整套的商业集成方案。但到目前为止很少有一家高校，完全依赖于外部商业解决方案来进行下一代数字校园建设。

在国内校级统一信息系统的概念是由清华大学最先提出，主要内容就是下一代数字校园建设的具体发展方向。这个理念完全打破了之前高校信息化建设的传统建设方法，从全局的角度去考虑高校信息化建设，从高校全局业务的角度去考虑问题引入了 IT 治理、顶层设计等内容[3]。

基于 SOA 的高校信息化研究最近几年进行的比较活跃，有些高校开始在这方面已经投入了不少的资源。面向服务的架构（SOA）在过去的几年逐渐发展成为系统设计、开发和集成的优选方案。利用开放的标准和无处不在的互联网，SOA 成为由自包含的逻辑工作单元构成的可重用服务的前提。SOA 带来的承诺是能够使用通用的模式将服务快速地组装在一起，以满足业务需求。由此带来的结果是更高的业务敏捷性以及 IT 资源和资产的高效能利用。SOA 并非仅仅是技术选择，像 IT 治理、服务质量等都是构建完整 SOA 的重要因素。

作为 SOA 实施的重要工具之一，ESB 产品近年发展的非常迅速。目前，很多中间件厂商已经开始关注 ESB 产品，把实现 EAI 的技术选型放在了 ESB 方面，而且他们的集成产品所考虑的范围逐渐扩大到整个 SOA 领域。ESB 为中心的产品越来越受到广泛的应用，而且有些 ESB 厂商的功能已经超出了简单 ESB 的概念，他实际上是一种综合的集成解决方案。

开源 ESB 中的 Service Mix 和 Mule ESB 目前都通过提供 BPEL 引擎来支持 BPM。而原来提供 SOA 平台产品的软件厂商则认为 ESB 只是实现 SOA 的一种工具，ESB 产品在 SOA 软件平台中的一部分，而通常又把 BPM 作为另一个单独的产品。

到目前为止，还没有一个统一的被认可的 ESB 标准。JBI(Java 业务集成)是一个由 SUN 公司（目前被 Oracle 收购）提出的 ESB 集成标准，对它的一个实现即为基于 JBI 规范的 ESB。Service Mix ESB 和 JBoss ESB 就是典型的基于 JBI 规范的 ESB 产品，而 Mule ESB 也提供了对 JBI 的支持。但是，JBI 并没有得到一致的支持，在业界基本上不看好 JBI 的发展前景。

1.3 本文研究内容

在课题中首先介绍了高校在系统集成方面的背景，分别列出了一些系统集成技术，包括 EAI、SOA 以及 ESB。第二部分主要讲述了企业服务总线技术。这部分重点介绍 SOA 和 ESB 的相关的内容。本章还介绍了 ESB 技术以及 SOA 与 ESB 之间的联系、开源 ESB 技术以及 Mule 开源产品的原理和特点，进一步理解 ESB 架构实现方式。第三部分主要讲述企业应用集成设计模式相关的话题。在本章中介绍主流集成设计模式和基于 ESB 设计模式，为集成项目设计提供依据。第四章是本课题的重点，也是解决本课题具体问题的过程。首先从高校面临的系统集成问题开始展开，分析需要解决的具体问题。第五章开始拿出一个具体的全局业务为例，结合 ESB 设计模式设计出基于 ESB 的校级统一信息系统。第六部分是基于 Mule 的具体实现过程。本章首先详细介绍了具体业务相关的服务接口，并且通过 Mule 的灵活配置实现服务之间的互操作。第七部分是总结和展望本课题，重点介绍项目实施过程中碰到的问题和下一步需要解决的问题。

第二章　企业服务总线技术研究

2.1　面向服务架构概述

面向服务架构概念是在 1996 年 Gartner 公司描述实施企业“V 英文”的时候，第一次提出来的[5]。刚开始 SOA 并没有受到广泛的关注，直到 XML 语言的出现及发展，以及 Web Services 等技术的发展，SOA 才从概念阶段逐渐转向于应用阶段。Web Services 最早由微软推动，2000 年时为更多公众所知。很快，IBM、Oracle、HP、SAP 和 Sun 等主要 IT 供应商加入了 SOA 的行列。当时，因为很多企业摸索着将自己的业务、系统及部门集成起来，整个 IT 界兴起 SOA 热潮。例如，2005 年 Gartner 公司说到：到 2008 年，SOA 将成为 80%开发项目的基础[6]。

SOA 概念已经出现了足够长的时间，到目前为止 SOA 并没有被广泛认可的定义，但无论如何解释，SOA 的核心思想就是增强灵活性。维基百科中对术语 SOA 的解释如下：SOA 是使用服务来满足软件用户需求的一种软件架构概念，在 SOA 环境中，网络上的节点以独立服务的形式将自己的资源开放给网络上其他参与者，其他参与者按一种标准的方式使用资源。这些服务就是各种业务细节的集合，SOA 就是通过协调这些业务细节给业务流程赋予新的生命力。SOA 的特别之处在于把这些服务进行专门的设计，目的就是使得服务之间是独立的、松耦合的以及可重用的。例如一种服务它可以参与不同的业务流程中实现不同的任务。

SOA 最适合于复杂的大型分布式系统，尤其适合于软件系统变得越来越复杂，更多的系统牵涉进来的集成项目。在 SOA 的实践过程中，大型分布式系统往往不是由唯一的所有者掌控，而是由不同的多个部门管理着不同的系统。因此，如何处理不同的平台、进度、优先级以及权限等问题也是 SOA 实施的重要内容。绝大多数大型分布式系统都是异质的，而且集成得越多异质化程度就越高。SOA 就是解决这种情况的思想，它完全接受了大型分布式系统的异质性，因此，SOA 关心的是怎样合适的方式去处理异质化。

SOA 有三个重要的技术概念：服务、松耦合、互操作性 [7]。服务就是业务功能的 IT 体现，在服务起作用的业务领域中，服务应该代表着一项自足的功能，对应着真实世界的业务活动。大型系统集成项目而言，可伸缩性和容错性是提高

可维护性的关键。而且在大型分布式系统中，一些修改或故障对整个系统环境的影响尽可能限制在特定范围内。SOA 的松耦合很好的解决了这方面面临的问题。SOA 之前企业应用集成（EAI）中大量的使用了互操作的特性，在 SOA 中 ESB 是互操作性的具体实现方式，也是跨越多个系统实现业务功能的基础。

2.2 企业服务总线概述

2.2.1 ESB 概念

ESB（Enterprise Service Bus，企业服务总线）到目前为止没有统一的定义，将来也不会出现统一的定义，因为ESB是根据不同的使用者有不同的意义。但很明显，不管ESB使用在什么环境，达到互操作性的目的都是一样的。它是传统中间件技术与XML、Web服务等技术结合的分布式中间件系统，用来面向服务方式的IT集成[7]。ESB又是一种具备Web Service 能力的基础设施，作为松耦合或解耦合的几个业务组件之间的中介，在集成环境中进行智能指导。ESB提供了开放的、基于标准的消息机制，通过标准适配器和接口，提供粗粒度应用服务与其他组件之间的互操作，能满足分布式异构系统的集成需求。除了这些还可以根据ESB的特性来进一步了解其概念。

根据维基百科的ESB定义，ESB有如下特性[8]：

1）它是面向服务架构的实现。

2）它通常是与操作系统和编程语言无关的；例如，它可以在Java和.Net应用程序之间工作。

3）它使用XML（可扩展标识语言）作为标准通信语言。

4）它支持Web服务标准。

5）它支持消息传递（同步、异步、点对点、发布-订阅）。

6）它包含基于标准的适配器（如J2C/JCA），用于集成传统系统。

7）它包含对服务编制（orchestration）和编排（choreography）的支持。

8）它包含智能、基于内容的路由服务（itenerary路由）。

9）它包含标准安全模型，用于ESB的认证、授权和审计。

10）它包含转换服务（通常是使用XSLT），在发送应用和接收应用之间转换格式，简化数据格式和值的转换。

11）它包含基于模式（schema）的验证，用于发送和接收消息。

12）它可以统一应用业务规则，充实其它来源的消息，分拆和组合多个消息，以及处理异常。

13）它可以条件路由，或基于非集中策略的消息转换，即不需要集中规则引擎。

14）它可监视不同SLA（服务级别合约）的消息响应门限，以及在SLA中定义的其它特性。

15）它（常常）简化“服务类别”，向更高或更低优先级用户做出适当的响应。

16）它支持队列，在应用临时不可用时用来保存消息。它由（地理）分布式环境中的选择性部署应用适配器组成。

2.2.2 ESB 功能

ESB的核心功能可以归纳为以下几点[9]：

1）地址透明化：通过ESB连接的服务消费者和服务提供者中，服务消费者不必知道服务提供者的精确位置。这意味着服务消费者与提供者是完全独立的，服务提供者的改动不会影响服务消费者。

2）传输中协议转化：ESB转化进入的传输协议，并把它转发到相应的出口中去。

3）消息转换：ESB把进入的消息格式转换为各种需要的消息格式。

4）消息路由：在复杂的集成项目中，很多应用系统都要把消息送到目标应用系统中去，ESB通过消息路由功能完成了应用系统间的消息传输逻辑和规则。

5）消息强化：为了正确的找到目标应用系统，ESB对传输消息进行额外的处理，也就是对传输消息进行基于数据库的客户端检验。与消息转换不一样的部分是消息转换是处理已经可用的输入消息，而消息强化是处理必须检索的外来数据。

6）安全：通常ESB将会处理关键性的业务，而且会涉及到很多应用系统。ESB是通过验证和授权等方法来提高安全性。

7）监视和管理：ESB在大型集成系统中处于核心位置，管理和维护的重要性是可想而知的。ESB自身提供了管理队列、监视消息大小以及监视消息吞吐量等功能。

2.2.3 ESB 的优势

本质上，ESB就是解决集成问题的技术产品。如果不使用ESB产品，而使用其他的集成产品，例如EAI，会怎么样呢？都知道传统EAI是点对点的集成架构方式。这种架构中，只能用手动来完成系统之间的接口开发和维护，在复杂的大型分布式系统中是其维护成本非常庞大。而且其架构本身的原因，可扩展性方面也受到很多的挑战。更何况，以业务启动为主的现代组织结构中，能满足千变万化的业务需求是选择集成产品的关键。ESB产品正好解决了了，上述EAI所碰到的各种问题，虽然目前还存在一些问题，但其灵活性方面的特点足以吸引很多集成项目相关的实施工作。

ESB的另一个优势在于，它很好的处理各种不同技术和协议之间的异质性问题。ESB将提供一种适配器技术，很容易解决IT环境中的异质性问题。

最后一种优势是它可以节省成本。前面也提到，与点对点的EAI架构比起来维护和管理成本会很少，而且由ESB产品集成的系统，能帮助组织机构灵活的应对目前面临的挑战，最终给组织带来很好的效益。

2.2.4 在 JavaEE 中使用 ESB

JMS（Java Message Service）作为消息的一种标准，在很多集成架构中使用。本文的焦点是开源ESB，而开源的ESB是基于JMS和其他的Java技术来构建的。本节将简单介绍ESB在JavaEE环境中的实现。

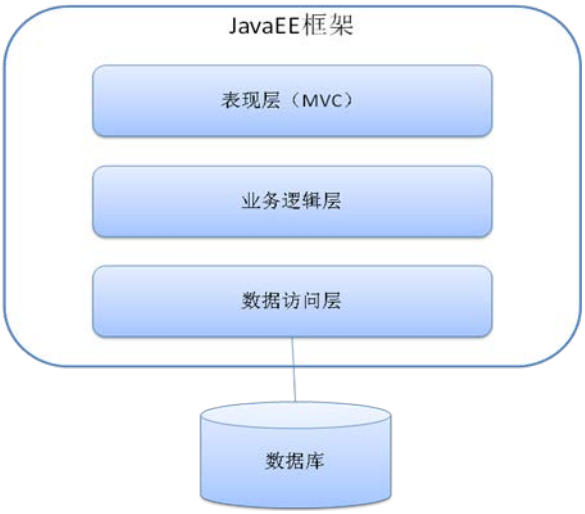

图2.1 JavaEE架构图

图2.1是JavaEE的三层架构图。三层架构把应用逻辑分成三个部分，不仅促进了应用程序的可伸缩性，而且加强了可维护性。这种方式的应用程序可以归类为孤立的应用程序，也就是它很难与其他应用程序进行交流。

为了与其他系统交互，在JavaEE三层架构中稍作修改，也就是增加集成层。这个层主要负责与其他系统交互。图2.2是在高校迎新系统中增加集成层后的架构图。图中可知，集成层和三个后端系统之间是直接链接的。这种连接方式看起来简单，但每个后端系统与集成层通讯时，其交互逻辑均不同。在这种情况下集成逻辑跟三个后端系统有关，也就是教务系统与迎新系统集成时，根据教务系统的通讯协议或消息格式来确定集成层的逻辑。但是有的时候，部分需要集成的应用系统与高校迎新系统没有必要进行交互。在这种连接方式中，不管需不需要一旦与集成层连接就进行集成逻辑，也就是消费很多不必要的数据交互。

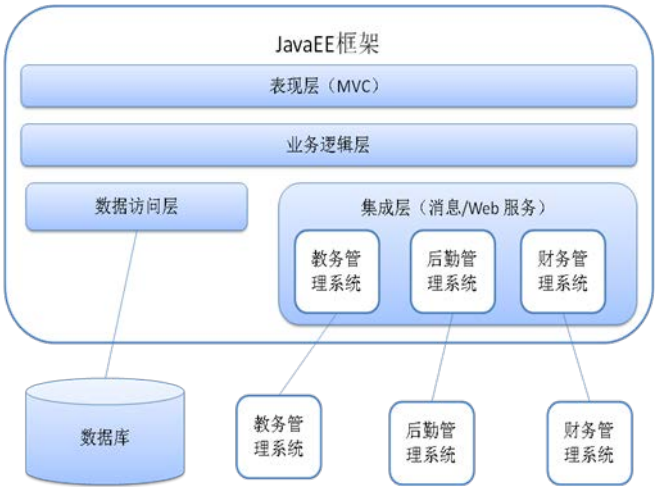

图 2.1 高校迎新系统架构

图2.3是使用ESB后的高校迎新系统架构图。在这种连接方式下三个后端系统的集成逻辑由ESB来负责解决，而集成层与ESB进行交互。这种连接方式的最大优点是把集成逻辑从应用系统中分离出来，它不再是应用系统的一部分。因为ESB专门提供集成功能，其他的后端系统专注于自己的应用逻辑。

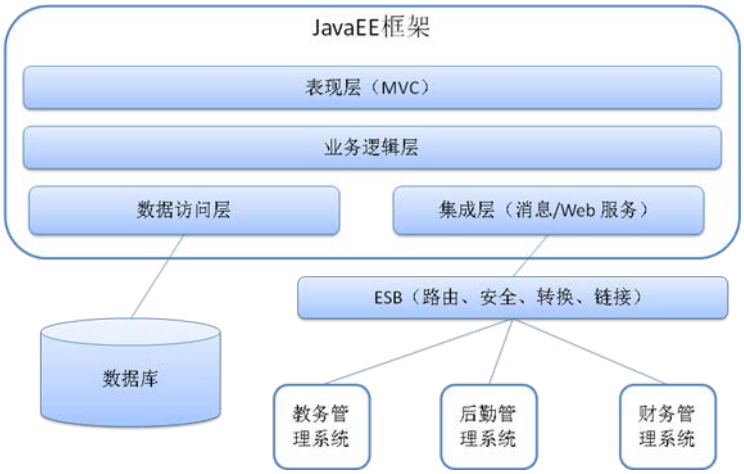

图 2.2 高校迎新系统中 ESB 的使用

图2.3是采用ESB把三个后端系统集成而成的高校迎新系统，这种设计的优势在于非常明确每个后端系统的在高校迎新系统中的必要性，也就是达到了系统集成的松耦合性目的。

2.2.5 增强 ESB 功能的技术介绍

Spring技术：Spring是一种组件框架技术，容易与POJO类一起工作。它的主要功能是依赖注入和控制反转。基于Mule的开发中部件逻辑的开发通常采用Spring的框架，因为Spring框架与Mule很好的结合在了一起，使用起来

非常方便。

JiBX：是XML-to-Java 的映射框架，它能把消息格式（XML）转换为POJO类对象，而且也可以把Java对象转换为XML。

ActiveMQ：是ESB消息代理的重要部分。消息代理一般提供异步消息交换和可靠传递等功能。在本文中通过该技术主要实现JMS代理。

2.3 开源 ESB 研究

2.3.1 开源 ESB 概述

开源软件是指免费获取该使用许可和源代码的软件产品，但针对该产品的服务和技术支持不一定会很全面，甚至很难找到相关的技术文档。而ESB是由一些商业公司提供开源ESB产品的比较多，例如Mule Source和IONA科技等公司。这些公司提供着24小时不间断的技术支持和丰富的技术文档。

开源软件通常是软件爱好者在业余时间里开发出来的产品，所以缺乏质量保证措施和发布规律。但是一些开源ESB软件是拥有着很优秀的bug跟踪系统、单元测试以及交互备份机制。除此之外，很多专职开发团队开发着开源ESB产品，所以在质量保证和软件发布等方面不会受太大的影响。

开源软件的又一个缺点是可用开发工具比较少。从最初开始，开源软件的优势不在于其开发工具上，这方面在ESB开源产品中也同样，很少有可用的开发工具。但是，最近开源ESB在可视化开发工具方面发展的非常迅速，比如在Netbeans中可以集成Open ESB开发工具了。

2.3.2 主流开源 ESB 产品介绍

开放标准和像JMS、JCA、XML、JBI、SOAP等Java规范的大量涌现推动了开源ESB产品的发展。很多对Java规范的开源项目，例如JBI和JMS，这些项目的实施是最终开源ESB产品出现的根源。其实，在这些开源项目的执行过程中出现了非常多的与ESB相关的产品，可是在这里只列出在集成市场上被受广泛关注的几种产品。

Mule ESB：是基于Java的轻量级ESB产品，是一种允许开发者更方便的、

更快的方式实现应用系统之间连接的集成平台，能实现应用系统之间的数据交换。Mule ESB能集成不同技术实现的应用系统，这些技术可以是JMS、 Web Services、JDBC、HTTP等等。Mule提供20多种的传输协议，并集成了大多数集成项目，包括Spring、ActiveMQ、Joram、CXF、Axis以及Drools。Mule并不是基于JBI规范的架构，它把重点放在提高开发效率的灵活性和轻量级的架构上。

Apache ServiceMix：是基于Java Business Integration（JBI）规范构建的，是在Apache的许可中发布的开源产品。而JBI的目的是使需要集成的组建和服务以独立的方式存在，让用户和集成商达到即插即用的效果。集成产品一直坚持JBI规范的原因是能够构建所有基于JBI规范产品部署的JBI组件。2007年9月，Apache ServiceMix已经成为Apache的顶级项目，该产品已经整合了很多其他的Apache产品。

OPEN ESB：是Sun公司（目前被Oracle公司收购）发起的ESB产品，属于Java.net项目。跟Apache ServiceMix一样，Open ESB也是基于JBI规范实现的产品。其实Open ESB的功能跟ServiceMix很相似，但一个重要的区别是：Open ESB是把焦点放在Glassfish应用服务上，而ServiceMix是更适合部署在Apache Geronimo应用服务或JBoss 应用服务上。Open ESB与其他ESB产品的另一个重要的区别是在支持的开发工具上OpenESB优于其他产品[9]。

Apache Synapse：本质上，Synapse是一个Web服务仲裁框架，是构建在Apache Axis2之上的。它与其他的ESB产品不同，关注的是提供一些路由、转换、消息验证以及基于Web服务和XML标准的注册。Synapse除了提供核心ESB功能外，还能提供综合的Web服务标准，比如WS-Addressing、WS-Security、WS-Policy以及WS- Reliable Messaging。

JBoss ESB：该产品被受广泛关注是因为JBoss公司的应用服务器和开源框架的出名。在企业级集成领域，JBoss提供了称为JBossMQ的JMS供应者和叫规则引擎的JBoss Rule。JBoss公司的集成架构中，除了JBoss ESB产品以外，还有其他一些JBoss产品，比如jBPM等[9]。

除了以上五种受广泛关注的开源ESB产品之外还有，基于Apache Synapse产品的WSO2和定位于EAI的OpenAdaptor产品。

2.3.3 选择Mule产品的意义

选择开源ESB产品的依据有很多因素。第一个依据是非常明显也是最重要的依据，也就是能满足前面2.2.2节中提到的ESB核心功能。还有一个比较重要的选择依据是有没有充足的可用文档。有没有高质量的文档往往是开发者们非常关注的问题。另一个选择的标准就是该产品在开源ESB领域有没有市场前景。在使用开源ESB产品时会碰到很多缺陷和额外的需求，因此在开源产品的讨论社区中得到解决缺陷的支持和帮助是非常必要的。在选择开源ESB时还有一个需要考虑的因素就是灵活性，也就是是否适合在客户的具体IT环境中灵活的完成集成任务。还有一个因素需要提出来的是有没有给开发者提供方便的IDE工具。

下面以几个方面讨论选择Mule ESB的意义。

1）ESB的核心功能：Mule ESB很好的满足所有ESB核心功能。

2）文档质量：Mule ESB是所有开源ESB产品中支持文档最全的ESB产品。

3）市场前景：在开源ESB领域，集成商的数量一直保持着增长的势头。Mule一直处在前列，并受广泛的关注，因为很多集成商的产品在很大程度上都依赖者该产品。

4）开发和交流的活跃度：在开发和交流的活跃情况方面，开源ESB的很多产品都比较不错。Mule通过Mule Source提供非常活跃的开发和交流的支持平台。

5）灵活性：大部分开源ESB产品都提供灵活的配置方式，Mule ESB也是不例外。

6）传输协议和链接方式：很多开源ESB产品都提供者大部分传输协议和链接支持，Mule ESB也是提供着广泛的链接和开源产品的支持。

2.4 Mule介绍

2.4.1 Mule组件介绍

Mule是由很多组件来构成的体系结构，这些组件相互结合在一起能提供

ESB的所有功能。图2.4展示着最基本的Mule结构概念。该图表明了主要的Mule组件把一个进入的消息传输到目标应用的过程。

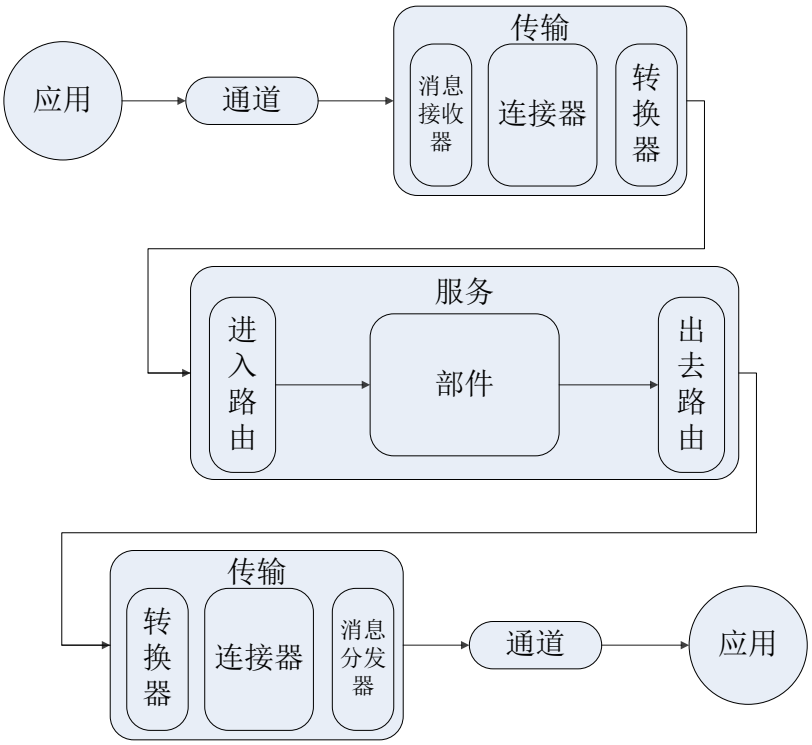

图 2.3 Mule 消息传输组件图

应用（Application）：需要集成的应用，可以是任何一种遗留系统，一种.NET应用或者是一种J2EE应用，甚至是其他的Mule实例。

通道（Channel）：通道提供了外部应用（Application）与Mule交流和链接的方法，它的功能就是把Mule和服务（Service）连接在一起。

消息接收器：这个部件就是从某些通道（Channel）接受消息，Mule提供了支持很多满足标准和技术的接收器。

连接器：连接器能识别怎么发送和怎么接受从通道发过来的消息，消息接收器和消息分发器也是连接器的一部分。

转换器：该组件把一种数据格式转换成另一种数据格式。

进站路由（Inbound router）：从通道（Channel）过来的消息是通过该组件才能确定下一步该做什么。

服务部件：该组件是Mule架构中的程序逻辑部分，与其他组件无关，能完成集成逻辑的任务。它可以是POJO、Groovy Script、REST Service和BPM等等。

出站路由：（Outbound router）：该组件与进入路由的功能差不多，出站

路由只是决定着经过程序逻辑的消息该往何处走。

消息分发器：该组件的功能与消息接收器相反，它知道怎样把信息发送到特定的通道中去。

下面几节将介绍Mule架构中起重要作用的概念组件。

2.4.2 Mule 端点

通道（Channel）提供了外部应用与Mule进行交流的依据，并且实现了与其他组件的连接。但是应用（Application）与通道（Channel）的连接还得需要一个方式，能起到该作用的叫做端点（endpoint）。实际上通道、连接器、发送者、接受者共同工作的结果就是端点。

2.4.3 转换器

如果收到的消息是从JMS队列传过来的或者是从邮件形式传过来的，能想象出所收到的消息格式是完全不同的。Mule对进出的消息使用了非常合理的默认的转换方式。当通过JMS收到消息时，Mule采用JMSMessage来自动转换，比如收到的是TextMessage时，它将转换成String格式， ObjectMessage是转换成Object，如果是其他的JMS消息格式也可以转换成相应的Java对象。这些操作都是自动完成的，不需要任何设置。当然，如果定制需要的转换操作，必须进行额外的转换设置，但是一旦进行特定的设置默认设置就无效了。

2.4.4 路由器

在前面的图2.4中可以看到路由器分为进站路由器（Inbound Router）和出站路由器（Outbound Router）两种形式。而且在图中还可以看到进站路由是应用于消息转换以后，而出站路由是应用于消息转换之前。路由器的功能就是确定消息接收后该转发的方向。在接受的消息当中进站路由器能过滤出特定的消息类型，例如它可以只让包含String类型或符合某种表示方式的消息通过。同样出站路由器也能过滤特定的消息类型。

2.4.5 服务部件

值得注意的是服务部件与Mule本身的代码无关，它可以是POJO、Spring bean、Java bean或者是拥有业务逻辑的Web service。Mule甚至还能提供称为云连接器（Cloud Connector）的服务部件，它能连接任何基于云计算的服务和应用。Mule帮助集成开发者管理这些服务部件，Mule是把服务部件绑定在配置文件中，使服务部件显露为服务的形式。

第三章　基于ESB的系统集成设计模式

3.1　系统集成设计模式介绍

设计模式是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结。使用设计模式的主要目的是为了更好地获得可重用性，使开发的软件系统更易于理解，同时还能保证系统体系结构的正确性和代码的可靠性[10]。设计模式在应用架构和设计领域方面被众所周知，尤其是GoF设计模式对每个面向对象开发者来说影响比较深刻。

大多数的开发者有一种倾向，就是想办法尽可能的忽略软件设计这一步骤，但在集成开发领域这一步骤起着决定性作用。系统集成专家设计集成方案时有可能把设计模式已经考虑进去了，但是这种实现方式不一定是符合系统集成的设计模式。目前比较常用的系统集成模式可以归纳为以下四种[11]：

文件传输：让每个应用生成共享数据文件，供其他应用使用，并使用其他应用生成的共享数据文件。

共享数据库：让应用把要共享的数据存储在一个公共数据库中。

远程过程调用：让每个应用公开提供自己的一些过程，使它们能够被远程调用，应用通过调用这些过程来执行操作并交换数据。

消息传递：让每个应用连接到一个公共的消息传递系统上，并通过消息来交换数据和调用行为。

选择系统集成模式的原则是根据具体的集成需求，灵活的考虑每种系统集成模式的优缺点，在每个集成环节中尽可能发挥相应的集成优势。消息传递的集成模式为集成标准提供了很好的平衡点，也是目前发展最快的集成模式之一。在系统集成过程中消息传递是受很多传递因素的依赖，而ESB恰恰提供了消息传递所需要的所有的功能。ESB产品的设计主要采用了各种消息传递的系统集成模式，例如消息路由模式、消息转换模式等等。

3.2　基于ESB的系统集成设计模式

为了更好的说明基于ESB的系统集成设计模式，在图3.1中展示了简单的

消息传输过程[11]。在图中通过6种消息模式实现了比较基本的消息传递过程，而这些消息模式就是下一步要讨论的ESB集成设计模式。

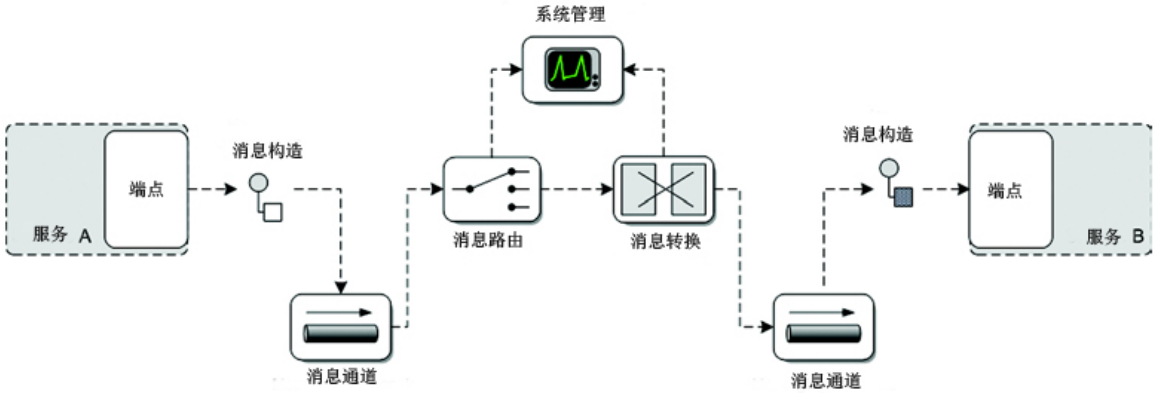

图 3.1　消息传输过程图

其实图3.1中所展示的消息传输过程与图2.4中所描述的基于Mule的消息传输很相似。这说明基于ESB的系统集成模式就是基于消息传输的系统集成模式，在ESB的功能中已经包含了很多系统集成的设计模式。图3.1中所看到的消息传输是从服务A消息端点（Message endpoint）开始的。这个端点将分发所构造的消息（Message construction）到消息通道（Message channel）中去，消息通道往往是JMS队列、HTTP URI或者是文件系统。该消息接下来将通过另一个消息通道被路由和转换到下一个端点的服务B中。在这一过程中消息传输被系统管理模式所监控，以便进行系统集成管理。

消息端点：大多数系统没有提供与ESB交互的内置功能。相反，这种交互必须有一层代码，这些代码不仅知道应用如何工作，还了解ESB如何工作，从而能够在应用和ESB之间能协同工作。这种能让它们协同工作的代码就是消息端点，支持应用发送和接受消息。典型的端点模式有幂等接收器（Idempotent Receiver）和选择式消费者（Selective Consumer）两种。

消息构造：消息是一种数据包，为了传送数据，应用必须把数据分解为一个或多个数据包，把每个数据包包装为消息。在接收端，接受者接收到消息后，必须从中取出数据并进行处理。该模式主要描述消息的类型和消息的元素。典型的模式是相互关系标示符（Correlation Identifier）和文件消息（Document Message）。

消息通道：消息通道是一种虚拟管道，是连接消息发送者和接受者。它提供了一种消息元素之间进行交换的方法。典型的模式有点对点通道

（Point-to-Point Channel）和被担保的投递（Guaranteed Delivery)。

消息路由：在实际的复杂集成应用中消息到达最终目标之前可能经过许多通道。路由器将决定消息在复杂通道拓扑结构中的传递方向，把消息正确地发送给目标接受者。典型的模式有基于内容的路由器（Content-Based Router）和聚合器（Aggregator)。

消息转换：在集成实践中各种应用可能对相同的概念数据采用了不同的数据格式，而且发送者和接受者希望的数据格式很难一致。消息转换器就是为了解决上述矛盾而出现的一种过滤器，是把一种消息格式转换为另一种消息格式。典型的模式有消息翻译器（Message Translator）和内容采集器（Content Enricher)。

系统管理:该模式就是处理监控和功能管理。典型的有总线控制(Control Bus）和线缆窃听（Wire Tap)。

第四章　基于 ESB 的校级统一信息系统需求分析

4.1　高校信息系统集成面临的问题

大部分高校已经意思到“信息孤岛”现象的严重性，但是到目前为止没有一个高校明确的提出适合自身的整体解决方案。高校信息系统集成已经持续了很长的一段时间，但是对很多高校而言实际操作起来还是面临着众多的挑战。更何况通过近十多年的发展，每个高校都有自身的信息化发展特点，都存在着独有的问题。一些普遍存在的问题可以归纳为以下几种：

1）没有决心以战略的高度去考虑系统集成问题。经过多年的实践已经知道，高校信息化只有整体的角度去考虑才能解决根本的问题，也是今后大部分高校信息系统集成要走的路。但是这种战略性的改动影响面非常广，处理不妥当后果非常严重。不了解学校信息化战略的紧迫性和敏感性，没有充分的准备和坚定的决心，实施操作将会遇到很多阻碍。虽然很多高校已经建立了校级别的信息化领导小组，以副校长兼任CIO等管理模式，但是具体执行的时候出现很多无法解决的问题。

2）没有充分认识目前所处的信息化现状。很多高校不知道自己在信息化发展中所处的位置，没有充分的认识到自身独有的问题。这种现象直接导致无法选择适合自身的系统集成解决方案。有的高校信息化程度还没有达到系统集成的条件就开始考虑这些问题，有的高校不考虑自身的实际情况就想模仿其他高校的集成方案。

3）对集成方案的选择上出现很多分歧。不断出现的集成技术和集成方案误导高校集成方案的选择。集成技术已经发展了二十多年，其实现技术非常丰富。如何选择适合自身的技术来实现信息系统集成已经逐渐成为一件非常复杂而困难的事情。因为它不只是简单购买一两个信息系统就解决的问题，更确切的说信息系统集成是买不到的产品。高校信息系统集成是逐渐实施的过程，需要耐心地去不断完善的过程。

4）对集成技术的掌握程度不够高。因高校自身的特点，高校虽然拥有着很强的系统集成理论基础，但是却很少有强大的系统集成实施团队，对信息

系统集成的实施经验也不够丰富。目前，大部分高校是由集成商来负责实施学校的信息系统集成项目，但是学校信息化部门的参与是不能忽视的。因为集成项目是长期性的工作，需要与集成商进行长时间的沟通和协调，甚至有时候只有信息化部门主导才能解决的问题。

4.2 校级统一信息系统需求分析

校级统一信息系统是学校统一组织建设的全局性信息系统，能够支撑学校的各种跨部门业务运转。校级统一信息系统具有六个核心要素，即：整合的基础设施、整合的数据、整合的用户、整合的应用、整合的权限和整合的流程[2]。校级统一信息系统建设思想很好的解决了目前大部分高校在信息系统集成所面临的问题，为学校信息化的可持续发展提供了有效的手段。

校级统一信息系统通过解决集中建设与分散建设的冲突，处理统一建设与组织结构分割的矛盾；通过进行跨组织边界的流程整合、优化和统一，做好数据标准和管理制度的统一；通过改变信息获取方法，提高师生员工应用信息系统处理业务的素质和能力；通过新老系统之间的关系协调，将原有的系统纳入到校级统一信息系统中。

校级统一信息系统不是单靠技术细节就能解决的问题，它是一种IT治理方案，是未来高校信息化建设要参考的发展模式，是学校层面上做出的重要战略决策。建设校级统一信息系统可能会导致学校的体制改革，需要完善管理模式，理顺管理流程，重组管理机构与队伍，统一组织实施等等，这不是一朝一夕能够完成的事情，需要长期的努力。因此必须结合学校的发展定位，考虑学校的治理结构，从全局出发，站在整体的高度考虑校级统一信息系统的建设问题[12]。

4.2.1 校级统一信息系统需求分析

校级统一信息系统的主要建设目标是通过全局业务的建设，进一步提升信息化对高校业务的价值。全局业务是高校中的某项业务流程，需要多个智能领域的协同工作才能完成[13]。从技术角度来说，凡是要通过2个以上系统

发生数据交换或流程互操作来实现的某项业务都叫全局业务[13]。

高校全局业务的建设包括以下内容[13]：

1）迎新系统：基于数据共享平台实现在迎接新生入校的整个流程中，各相关部门间的数据整合和管理功能。

2）离校系统：基于数据共享平台实现与各部门系统之间的无缝集成，实现各审核节点自动审核或处理。

3）资产综合管理系统：所涉及的部门比较繁杂，通过统一的入口实现资产管理的标准化，实现与财务数据的一致。

4）人力资源管理系统：该系统是面向全校涉及与人力资源有关的服务平台，从学校的全局管理角度出发，充分考虑了各部门与人力资源之间的联系。

5）科研管理全局系统：科研管理流程是高校的核心业务，该系统提供统一的科研管理标准，实现与财务、人力资源、校友、教学以及综合门户等系统之间的联系，构成复杂的业务流程。

6)统计与决策管理系统:基于各种全局业务数据的跨领域数据共享平台，实现全校范围内的各种数据统计。

高校全局业务的建设需要三步走战略：

1)学校不同领域之间的数据同步阶段:业务系统需要其他域提供的数据，以实现数据完整、实时等目标。需要确定各类数据的权威源，并实现权威源和各系统之间的数据交换。这期间的主要工作是实现基础设施整合、数据整合、应用系统整合、以及用户整合。

2）学校不同领域之间业务集成阶段：在数据交换、数据同步、应用集成的共同参与下，实现复杂的全局业务流程。本阶段的主要工作是流程整合。

3）学校不同领域之间数据共享阶段：基于数据中心，实现全局的数据共享和利用。这期间的主要工作是权限整合和业务规则整合。

以上的前两个步骤，为了考虑风险，可以先在小范围内实施。比如对一些学校的关键业务影响比较小的系统开始部分地进行集成，逐渐扩大集成范围，在这一过程中逐步完善基础设施、用户、数据等整合工作。第三个步骤必须以前两个步骤为基础，尤其需要业务流程整合的进一步完善。

4.2.2 基于 SOA 的校级统一信息系统

为了满足校级统一信息系统建设要求，我们需要一种新的体系结构，它既能有效地利用现有的IT 基础设施，又具有足够的灵活性和适应性，能与不断变化的业务流程和业务模型保持一致。SOA无疑是一种非常合理架构选择，在第二章中已经介绍到SOA注重的是可重用性和灵活性，它的目的就是尽量把现有的业务系统通过高质量的服务化和组件化过程来完成我们需要的业务流程整合。

通过SOA的思想实现校级统一信息系统的过程中，SOA的能力成熟度模型

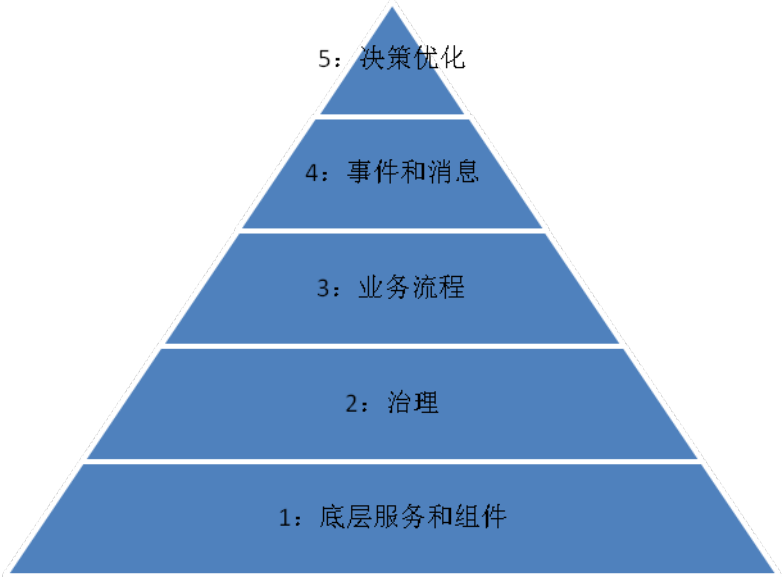

图 4.1 SOA 能力成熟度模型

很有借鉴意义[14]。图4.1是SOA的能力成熟度模型，分为5个层级[14]。每个层都依赖处于下方的层级。第一和第二层对应校级统一信息系统实践步骤中的第一个步骤，也就是数据同步阶段。在SOA的成熟度模型中一、二层的关键工作在于搭建基础设置和IT治理。ESB的搭建工作是所有SOA实践的基础，也是确保服务是否通过通用的框架开发并带有相关的安全策略来进行交互的关键所在。在SOA成熟度模型中的IT治理阶段如果从技术层面上讨论，主要内容是如何有效暴露，并发现可重用的服务，毕竟SOA是面向服务的架构，服务的有效利用将影响着整个集成工作。第三层对应的是校级统一信息系统实践步骤当中的第二个步骤，也就是业务流程集成阶段。在SOA的能力成熟度模型当中第三层的主要工作是构建更加粗粒度的组合服务。组合服务通常都需要结

合业务逻辑来决定如何从逻辑上更底层的服务组合在一起。第四和第五层对应是校级统一信息系统实践步骤中的第三个阶段，也就是数据共享阶段。SOA能力成熟度模式中属于第四层的消息和事件是SOA领域中的新鲜事物，也是今后的发展趋势。能更早的引入消息或事件驱动的架构是实现决策优化层（第五层）的基础。实时地监控所发生的事件，对于优化业务流程及事务性的决策是很重要的。

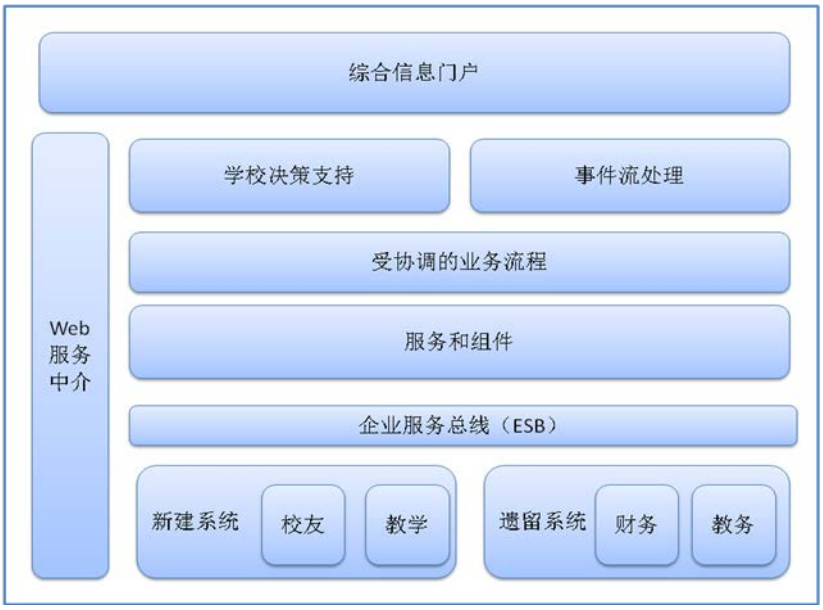

图 4.1　校级统一信息系统 SOA 架构图

图4.2是校级统一信息系统的SOA架构图[14]。本文主要针对SOA能力成熟模型的第一层以及与第二层之间的联系进行讨论，也就是如何通过ESB来实现SOA的基础设施，如何安全有效地实现SOA中的互操作功能。三层以上的内容超过了本文讨论的范围。

4.3　基于 ESB 校级统一信息系统需求分析

实现完整的跨域共享数据中心是完成校级统一信息系统的重要标志之一，而不同域之间的数据同步又是实现共享数据中心的基础。数据同步关注的是各种数据资源的元数据管理、数据流的逻辑处理、各种数据接口的处理、业务数据清洗等[15]，保证各个系统中所存储数据的唯一性和权威性。实现基于ESB的数据同步时需要遵守以下原则：

1）低耦合：由于数据同步需要与多个系统进行数据的对接，因此在设计

上必须考虑耦合性的问题。多个系统在进行系统集成的过程中不可能在短期内全部部署，而只能通过规划逐步与数据交换平台衔接上线，这样就需要在设计过程中尽量降低应用系统与共享数据中心间的耦合性，减少系统的数据依赖。

2）数据安全性：由于数据同步过程中的数据大部分是与业务相关的核心数据，并且也会涉及到个人隐私的数据，因此对于其安全性要求会很高。通过系统的多级权限控制以实现不同权限级别的人员只能访问其权限范围内的数据，以保证数据的保密性以及安全性。

3）保证数据传输性能：数据同步是共享数据中心的基础平台，所以该平台必须能支撑上万级的数据查询和并发事务的处理能力。而且更重要的是还得考虑今后业务改变所带来的各种系统压力。

4）数据完整性：数据同步将采用与各个应用系统完全不同的数据模型，这种数据模型为数据完整性提供了重要的依据，以保证共享数据的权威性。

通过数据同步技术的分析，发现ESB完全可以胜任数据同步阶段的基础设施搭建任务，ESB在灵活、传输、安全、协议转化以及管理等方面完全达到了所需要的标准。在接下来的章节以高校迎新系统的建设为例简单介绍基于开源ESB的校级统一信息系统的实现过程。

第五章　基于 ESB 的校级统一信息系统设计

高校迎新系统是校级统一信息系统的重要内容，也是学校全局业务的一个重要体现。高校迎新系统的实现过程中各部门的数据标准化和共享是最重要的目标之一。高校迎新系统通过新生数据的集中管理，以权限划分为依据实现各部门在最大范围内实现数据的共享。高校迎新系统以数据共享为基础实现数据的完全性、准确性和唯一性[16]。

5.1　设计流程和服务

高校迎新系统中报到的环节是所有迎新数据共享的重要体现，下面以新生报到为例设计迎新的流程。

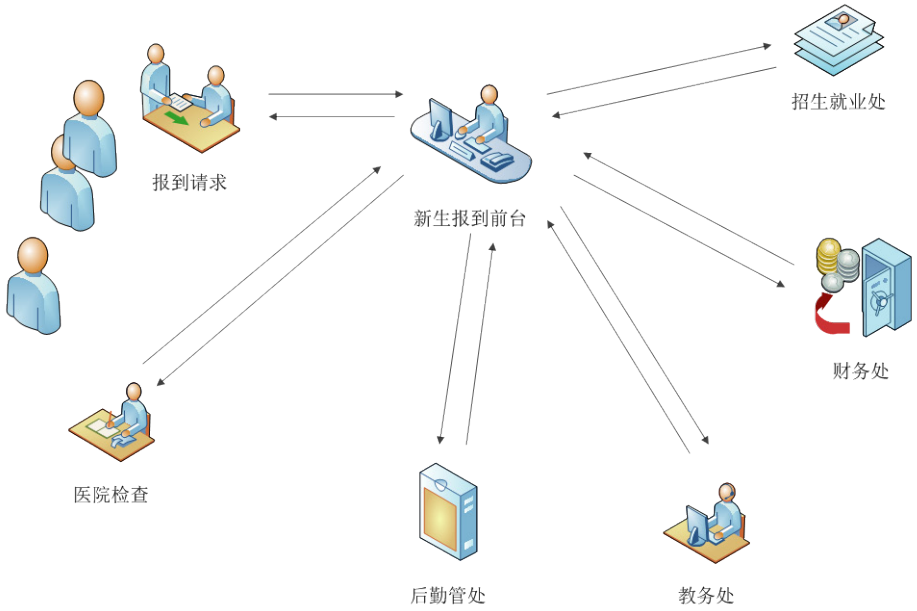

图 5.1　迎新流程图

学生完成校医院检疫和交学费步骤以后到新生报到前台进行报到。新生报到前台根据相关部门的业务处理数据进行审核。审核过程中迎新系统根据新生信息与招生系统、校医院管理系统、财务系统、教务系统以及后勤管理系统进行数据交换。报到完的学生可以进行下一步的入学操作，并根据自己的信息查询入学操作进行的情况。

新生报到的审核过程包含很多消息传递，比如新生报到前台请求招生系统获取招生信息，请求校医院检疫管理系统获取检疫情况，以及财务系统回

应新生交学费情况等等。下一步要做的工作就是调整这些消息结构，也就是重新设计这些消息让它们能够在服务和流程之间进行传递。在ESB中消息传递通常采用的XML格式，制作消息编排策略来满足消息之间的转换操作。

当新生申请报到时，新生报到接待员将根据新生的录取编号查询新生报到情况，这时候迎新系统将与招生系统、财务系统、校医院管理系统、后勤管理系统进行消息传递。下面更详细的说明与迎新系统进行审核操作时的一些服务，在关联的系统中暴露出了以下的关键服务。

录取信息服务:迎新系统根据新生的录取编号请求招生管理系统，招生管理系统查询到匹配的新生招生信息，在回应的消息中包含新生的录取状态以及新生的学号等关键信息。

班级信息服务：迎新系统根据新生的学号请求教务系统，教务系统查询匹配的新生教务信息，并在返回的消息中包含新生的专业、班级等关键信息。

缴费确认服务：迎新系统根据新生的学号请求财务系统，财务系统查询匹配的新生财务信息，并在返回的消息中包含新生的缴费信息。

检疫确认服务：迎新系统根据新生的学号请求校医院管理系统，校医院管理系统查询匹配的新生检疫情况，并在返回的消息中包含检疫结果信息。

入住信息确认服务：迎新系统根据新生的学号请求后勤管理系统，后勤管理系统查询匹配的新生寝室分配情况，并在返回的消息中包含寝室分配情况。

在应用系统和服务之间传递的消息会不同的方式存在，例如Java对象或者XML消息格式。一般情况下一所高校的信息系统不可能全都是Java开发的应用系统，有可能是基于.NET开发的应用系统或者是PHP。在这种情况下需要一种消息格式来统一传递工作，本文中将采用XML来统一消息格式。图5-2中显示了各个相关服务通过ESB与迎新系统互操作的过程。

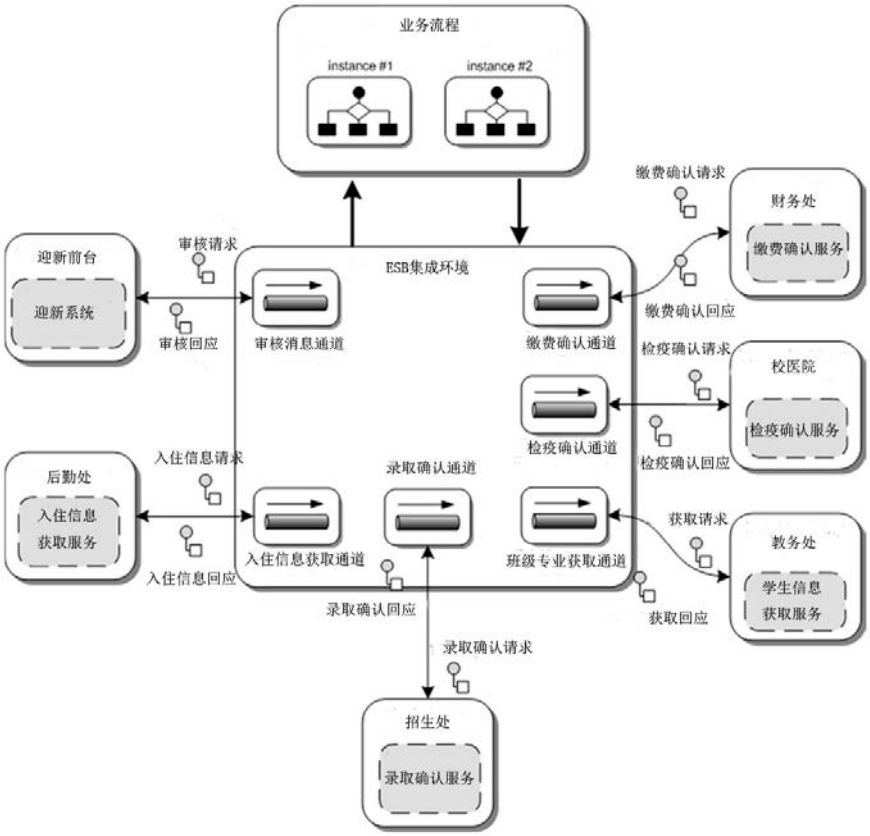

图 5.2 迎新系统 ESB 集成环境架构图

图5.2中有业务流程的概念。业务流程不是本文研究的范围，但是为了实现迎新系统的新生报到确认操作，不得不引入业务流程方面的工作。这些被暴露出来的服务最终是被业务流程使用，并完成相应的操作。业务流程负责处理逻辑业务，而ESB是负责搭建集成环境（服务和应用）。业务流程取代的是ESB的消息路由方面的工作。本文中采用jBPM开源业务流程开发工具来实现业务流程设计方面的工作。

图5.2中有很多服务给暴露了出来，这些服务在迎新报到的过程中被使用。业务流程通过消息通道获取招生确认请求消息后进行流程实例化，再把结果通过消息通道返回给迎新系统。

5.2 基于 ESB 的校级统一信息系统设计

基于ESB的系统集成实际上是基于消息传递技术的集成，本文中将采用在第三章已经介绍过的集成设计模式来设计迎新系统中新生报到确认操作过程。

5.2.1 设计消息通道

迎新系统的新生报到确认操作由很多的消息交换来组成。首先需要确定的是消息怎样传到目的地，比如招生系统或教务系统。在第四章中已经提到，消息传递方面的决定与消息通道模式有关。有两种消息通道模式在消息传递中经常使用，它们分别是点对点模式和发布/订阅模式。

由于迎新系统的新生报到操作中所需要的特定信息与特定的消息提供者有关。也就是迎新系统所发出的请求信息中有招生确认有关的，也有教务信息有关的，也有财务信息有关的，所以各相关系统只能处理与自己有关的消息内容，跟自己无关的消息就丢弃。点对点的消息通道模式非常适合这种场景，因为该模式能确保任何给定的消息只有一个接受者能消费。有时通道的接收者可以有多个，但是其中只有一个能成功的消费某个特定的消息。

5.2.2 设计消息构造

迎新系统发送消息时，往往希望能收到应答，以确认消息已经得到处理，并提供相应的结果。请求/应答的消息构造模式就是这种情况。请求一般是一个命令消息，而应答中包含了一个结果值或一个异常。在请求消息中指定返回地址，并告诉应答者使用哪个通道传送应答。比如与招生系统交换消息时，迎新系统请求招生信息查询，招生系统执行查询并在应答中返回查询结果。

5.2.3 设计消息路由

当迎新系统处理新生报到请求时，发送请求消息，这时候ESB需要路由器来处理这些消息正确的传递到目标系统。比如与招生确认有关的请求正确的传递到招生系统中，而与教务信息查询有关的请求正确的传递到教务系统中。基于内容的路由模式解决了迎新系统传递消息时需要处理的路由问题。基于内容的路由器首先检查消息的内容，并根据消息中的数据把消息路由到不同的通道上。

5.2.4 设计消息转换

迎新系统接收回应的消息时所需要的信息比较多而且复杂，而这些信息是通过招生系统、教务系统、财务系统、后勤系统、医院管理系统来获取。这些提供确认服务的系统只提供新生报到时必要的信息。内容扩充器是消息转换模式的一种，它主要把输入消息中的信息从外部数据源中获取数据。内容扩充器从数据源获得所需的数据后，就把这些数据附加到消息中，迎新系统所接收到的消息与新生报到所需的信息相符。

从教务系统中获取学生信息时学生信息获取服务所创建的消息有可能包含很多该学生相关的信息。但是很多情况下为了安全考虑只有教务系统才能查看学生的全部信息，其他系统不允许查看。内容过滤器模式是消息转换模式的一种，它的主要作用就是从结果消息中删除一些不必要的敏感信息。通过内容过滤模式后的迎新系统中，只显示与报道确认有关的一些必要信息。招生系统、财务系统、医院管理系统提供相应的确认回应消息时也是采用内容过滤模式来处理，减少不必要的信息提高通讯效率。

以上几种对于消息流的设计已经全面覆盖了基于ESB的设计范围，接下来的章节中将使用优秀的开源ESB产品Mule来实现已设计好的迎新系统的新生报到确认操作。

第六章　基于 Mule 校级统一信息系统实现

前面章节中采用基于ESB的集成设计模式，设计好了迎新系统在新生报到确认环节的操作过程。本章将采用Mule ESB实现这一过程。

6.1　Web 服务的实现

迎新系统的报到操作起始于招生信息的确认，也就是迎新系统请求招生确认服务来完成第一步操作。以下是与招生系统连接时所需的接口代码。

```
Public  interface  EnrolmentService  {
    Public  ConfirmEnrolment  confirmProcess(EnrolmentReq  request);
}
```

招生确认服务接口提供了一个方法，也就是提供招生确认请求以后返回招生确认信息。在该接口的实现类中需要实现的业务逻辑是把根据获得的确认请求信息从招生系统中获取符合条件的确认信息，并返回结果。

```
Public  class  EnrolmentServiceImpl  implements  EnrolmentService  {
    Public  ConfirmEnrolment  confirmProcess(EnrolmentReq  request) {
        if (request.getConfirm.equals(true))  {
            ConfirmEnrolment  confirm  =  new  ConfirmEnrolment();
            confirm.setConfirmId(new  Random().nextLong());
            confirm.setEnromentNum(request.getNum());
            return  confirm;
        }else{
            return  null;
        }
    }
}
```

招生确认服务是通过EnrollmentReq对象的confirm属性来判断，当查询结果是true时返回该学生的招生编号和招生确认ID。如果没有查到该学生将返回null信息。迎新系统中有一个判断代码，如果所得到的招生确认信息是null，直接提示该学生没有被录取的信息。

类似医院检疫确认服务的实现类代码如下。

```
Public  class  quarantineServiceImpl  implements  EnrolmentService  {
    Public  ConfirmQuarantine  confirmProcess(QuarantineReq  request) {
        ConfirmQuarantine  confirm  =  new  ConfirmQuarantine ();
        confirm.setConfirmId(new  Random().nextLong());
```

```
            confirm.setStudentNum(request.getNum());
            confirm.setResult(request.getResult);
            return  confirm;
            }
        }
    }
```

医院检疫确认的实现类根据学生录取编号查询学号，再根据学号查询学生检疫结果。返回的的确认信息中包含学生的学号和检疫结果。班级分配情况的确认、缴费情况的确认、宿舍分配结果获取等操作的实现代码与上面的实现类代码比较像是，不同的是返回的字段。

6.2 消息流实现

基于Mule的开发中大多数情况下使用XML格式的配置文件，这个配置文件是Mule开发的核心。Spring框架为Mule开发提供了非常方便的环境，本例中将使用大量的Spring标签来实现Mule配置。图6.1是Mule与jBPM结合工作的流程图，图中只有对关键的流程节点做了介绍，并没有深入的去分析流程实例。

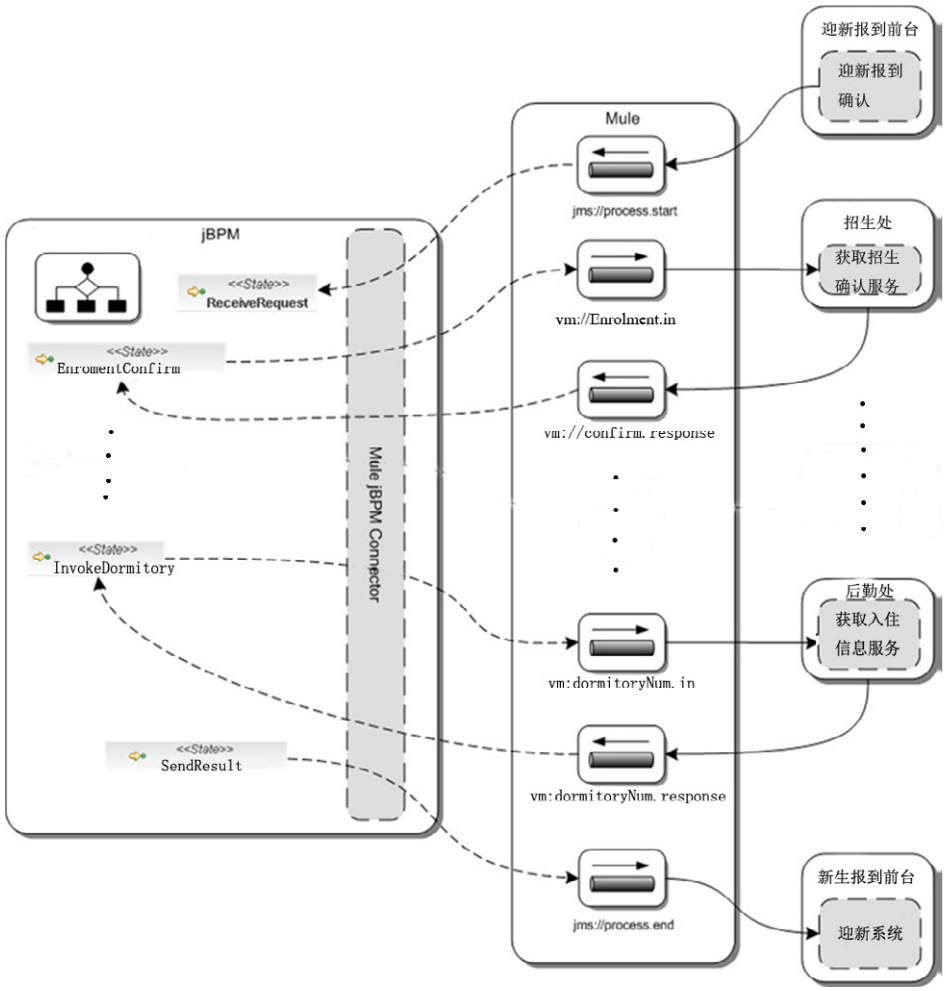

图6.1 Mule与jBPM结合工作的流程图

图6.1中显示了在新生报到确认操作中该业务流程如何通过Mule与关键服务进行互操作[9]。以下是基于Mule的迎新系统报到过程的实现配置文件mule-config.xml。

```
<mule>
    <spring:beans>①
        <spring:import resource="jbpm-beans.xml"/>
        <spring:import resource="register-beans.xml"/>
    </spring:beans>
    <spring:bean id="jbpm" class="org.mule.transport.bpm.jbpm.Jbpm"
    destroy-method="destroy">②
        <spring:property name="jbpmConfiguration">
            <spring:ref bean="jbpmConfig" />
        </spring:property>
    </spring:bean>
    <bpm:connector name="jBpmConnector" bpms-ref="jbpm" />③
    <jms:activemq-connector name="activeMQConnector"
```

```
brokerURL="tcp://localhost:61616"/>④
<custom-transformer name="XMLToConfirmRequest"
class="esb.util.framework.XMLToObjectTransformer">⑤
<spring:property name="targetClassName"
value="esb.Register.model.RegisterReq"/>
</custom-transformer>
<custom-transformer name="ObjectToXML"
class="esb.util.framework.ObjectToXMLTransformer"/>⑥
<bpm:endpoint name="ProcessEngine"
process="RegisterConfirmProcess" />⑦
<jms:endpoint name="registerRequest" queue="process.start"/>
<vm:endpoint name="EnrolmentIn" path="Enrolment.in"/>
<vm:endpoint name="ConfirmResponse" path="confirm.response"/>
<jms:endpoint name="registerConfirm" queue="process.end"/>
<model name="registerModel">
   <service name="ToBPMS">⑧
   <inbound>
      <inbound-endpoint ref="registerRequest"/>⑨
      <inbound-endpoint ref="ConfirmResponse"/>⑩
   </inbound>
      <outbound>
         <filtering-router>
            <outbound-endpoint ref="ProcessEngine"
               synchronous="false" />⑪
         </filtering-router>
      </outbound>
   </service>
   <service name="FromBPMS">
      <inbound>
         <inbound-endpoint ref="ProcessEngine"/>⑫
      </inbound>
      <outbound>
         <bpm:outbound-router>
            <outbound-endpoint ref="EnrolmentIn"/>⑬
            <outbound-endpoint ref="registerConfirm"/>⑭
         </bpm:outbound-router>
      </outbound>
   </service>
   <service name="Enrolment">
      <inbound>
         <inbound-endpoint ref="EnrolmentIn">
            <transformer ref=" XMLToConfirmRequest"/>⑮
         </inbound-endpoint>
      </inbound>
```

```
            <outbound>
                <pass-through-router>
                    <outbound-endpoint ref="ConfirmResponse">
                        <transformer ref="ObjectToXML"/>⑯
                    </outbound-endpoint>
                </pass-through-router>
            </outbound>
        </service>
    </model>
</mule>
```

第五章中讲到迎新系统的报到确认操作中将涉及到业务流程，所以在配置文件中首先导入了与业务流程相关的两个文件①。这些文件中将实现业务流程实例的具体步骤，该文件的内容超出了本文讨论的范围。为了实现Mule服务与流程之间的沟通，在配置文件中必须配置BPM连接器②。而且由于BPM的连接是有很多入口，所以还得配置bpms-ref属性③。jBPM的配置是通过Spring Bean 实现的，它使用了Mule传输框架的Jbpm类。配置文件中利用ActivMQ连接器来实现JMS消息传输④。本配置文件中有两个消息转换配置，分别是从XML转换到注册请求信息和对象转换到XML，其中在XML转换为注册请求对象时利用Spring的注入功能确定了目标类的名字⑤⑥。

由于前面已经完成了jBPM传输连接配置，现在可以定义一个端点与jBPM流程进行交流⑦。为了实现新生报到确认流程，在配置中进站消息和出站消息中分别定义ToBPM和FromBPM服务⑧。ToBPM定义了所有的发送的消息，包括初始的请求信息，而FromBPM服务中定义了所有接收的消息。在ToBPM服务中入站有两个端点分别是新生报到请求和返回确认端点⑨⑩，而出站端点是包含在路由过滤器中，过滤器采用了BPM引擎。这种过滤器可以理解为更复杂的基于内容的路由或过滤。注意到出站端点中把异步传输模式设置为false⑪。在FromBPM服务中入站端点使用了BPM引擎⑫，而出站端点分别是招生信息确认和报道确认端点，这两个出站端点在前面已经定义过⑬⑭。

Enrollment服务进站端点中调用了XML格式转化为ConfirmReq对象的转换器⑮，该转换器也是在前面已经定义了。Enrollment服务的出站路由中使用了outbound-pass-through-router标签指定了出站端点，而且出站时进行了Object到XML的格式转换⑯。

前面的配置文件中只显示了招生确认信息的获取操作，剩下的与教务、财务、

检疫、入住等信息的确认操作和招生确认操作非常相似，在这里不再继续讨论了。

第七章　总结和展望

7.1　总　结

本文重点讨论了基于开源ESB的校级统一信息系统实现过程。通过本次论文研究，作者对校级统一信息系统相关的认识提高了不少。校级统一信息系统在解决目前高校信息化建设面临的问题中起到非常积极的作用，是下一代数字校园建设的重点。开源SOA平台为校级统一信息系统的实施提供了有力的支持。本文的重要成果之一是把SOA思想融入到校级统一信息系统的实施之中，尤其是在开源SOA平台的有效利用和结合方面得到了重要的进展。其中本文重点讨论了ESB方面的技术。Mule ESB技术的应用过程中发现Mule技术资料非常全面，而且社区的交流也非常活跃，在具体实施过程中这些资源的有效地利用使我减轻了不少麻烦。本文还有一个重要的讨论就是基于设计模式的系统集成话题。基于消息流的系统集成是目前最先进的集成方法，也是大部分ESB产品的理论依据。如何有效地使用这些集成设计模式是基于ESB的集成项目成功的关键。本文以校级统一信息系统为依据具体分析了目前高校全局业务的需求，确定了一些需要整合的业务流程。其中重点研究了迎新系统的新生报到过程。按照SOA的开发步骤，首先暴露了迎新系统相关的各种Web服务，然后根据各种集成设计模式设计迎新系统中的消息传递过程，把相关的Web服务有效地连接起来。而且具体实施的时候采用了MuleESB架构，通过具体的开发认识到Mule技术非常丰富，除了能满足基本的ESB功能还能支持一些高级的功能，例如与业务流程框架的结合等等。

7.2　下一步工作

本文的研究范围比较杂，因为校级统一信息系统这个话题本身就是非常庞大，如果想把它具体的实施下去，那么所涉及的技术非常复杂，并不是短短几年就能完成的任务。所以本文只讨论了对学校关键业务不太重要的、影响比较小的迎新系统。在实施过程中逐渐认识到ESB只是SOA实现的一种工具而已，它只是帮助开发者更有效的完成系统集成问题。

其实ESB的实施在SOA项目中的地位很重要，但并非是SOA实施步骤当中的第

一步。本文虽然是关于ESB的研究，但更重要的是通过本次研究更详细的了解校级统一信息系统和开源SOA的架构。因为开源项目对于高校来所意义重大，尤其是很多地方高校来所。在大部分地方高校受到资源短缺影响的时候，开源软件的出现无疑是给它们带来的雪中送炭的效果。因为高校中的SOA项目实施是长期的过程，它需要漫长的投入，所以下一步的工作应该多放在基于开源SOA平台的学校小范围业务的实施上。

参考文献

[1] 刘希俭. 推进统一的企业信息系统建设[C] 2007 信息化推进大会. http://down.eiw.com.cn/Conference.

[2] 蒋东兴，宓詠，郭清顺 高校信息化发展现状与政策建议[J] 中国教育信息化 2009 （8）：27-30.

[3] 蒋东兴，王进展，袁芳，褚庆军 数字校园校级统一信息系统建设研究与实践[J] 中山大学学报 2009 48(z1):12-15.

[4] 张友生. 系统分析师技术指南[M] 清华大学出版社 2007 :188.

[5] 凌晓东. SOA 综述[J] 计算机应用与软件 2007 24(17):122-124.

[6] 王永强. 遭遇开源、SOA IT 规划方程式正在改写 2006[C] （9）:54-56.

[7] Nicolai M.Josuttis SOA 实践指南[M] 电子工业出版社 2008:16-18,21,276.

[8] 维基百科的 ESB 定义 http://en.wikipedia.org/wiki/Enterprise_service_bus

[9] Tijs Rademakers，Jos Dirksen Open Source ESB in Action[M] Manning Publications 2009:13，23-29.

[10] 杨少波，卢苇 J2EE 项目实训——UML 及设计模式[M] 清华大学出版社 2008:205-206

[11] Gregor Hohpe，Bobby Woolf 企业集成模式[M] 中国电力出版社 2006:28-29,

[12] 袁芳、蒋东兴、郭大勇 关于校级统一信息系统建设模式的探讨[M] 计算机工程与应用 2009 45:1-5.

[13] 茅维华 高校信息化中的全局业务 2009 年高校信息化经营论坛报告[C] http://net.sjtu.edu.cn/ReadDoc/szdx/doc.html.

[14] Jeff Davis 开源 SOA[M] 电子工业出版社 2010:24-25，

[15] 林怀恭，聂瑞华，罗辉琼，黄序鑫，马将 基于 ESB 的共享数据中心的研究与实现[J] 计算机应用与软件 2010:185-187.

[16] 尹世学，陈怀楚，黄卫卫，王英学 清华大学数字迎新系统的设计与实

现[J] 2006

http://www.cic.tsinghua.edu.cn/thcic/detail_30year.jsp?boardid=20306&seq=5024&pageno=3

[17] 冯培培,王辉 基于ESB技术的系统集成框架的研究[J] 通讯技术 2010 45（1）:195-197.

[18] 阚志刚，张润彤 基于Service Mix的SOA架构的研究与实现[J] 计算机工程与科学2009，31(4):153-156.

[19] 梅立军，付小龙，刘启新，沈锡臣 基于SOA的数据交换平台研究与实现[J] 计算机工程与设计 2006，27(19):3601-3603.

[20] 马文龙，余文利，廖建平 一种基于SOA的高校信息系统集成模型设计与实现 计算机时代 2010(2):41-43.

[21] 蒋东兴，郭大勇，罗念龙，刘启新 清华大学新一代数字校园建设规划与实践 厦门大学学报 2007，46(z2):173-178.

作者简介及在学期间所取得的科研成果

姓名	蔡珉官	**性别**	男	**出生日期**	1980.9.8
民族	朝鲜族	**出生地**	吉林省龙井市	**学位**	学士
职称	工程师	**学历**	大学本科		
工作经历	2003 年 7 月--2008 年 7 月，在延边大学信息化中心网络部工作，主要负责接入网管理、骨干网管理、计费系统管理、邮件系统管理等方面的工作； 2008 年 8 月至今在延边大学信息化中心研发规划部工作，主要负责软件开发，先后完成了延边大学继续教育学院教务系统以及延边大学校友录系统。				
联系方式					
著作与成就					

致　谢

通过本次论文研究，在各方面收获不浅，不仅在技术研究能力上，还是在实际工作当中都得到了显著地提高。非常感谢刘磊教授的关心和指导，如果没有刘磊教授的正确指点我可能走不到现在。每次碰到重要问题时都是教授给了我非常重要的建议，使我很快的摆脱困难。多次与教授交流的过程中学到了不少专业知识，更重要的还是一些研究问题的方法。这些方法不仅在写作过程中起到了非常重要的作用，而且在实际工作中帮了我不少的忙。我还要感谢所有任课教师，在研究生期间对我的教会和帮助。通过研究生期间的学习，不仅学到了专业知识，而且学会了研究和分析问题的方法。在此，我还要感谢金永灿老师，在百忙当中给我提供了很多高校信息化建设方面的资料和经验，为我的下一步工作起到了坚实的基础。同时还要感谢在论文写作期间对我提供帮助的其他所有人。