

中图分类号:

UDC:

学校代码: 10055

密级:

南开大学
硕士学位论文

中文题目: 异构信息服务平台在航空业的研究和应用

English title: The Research and Application on aircraft industry's Heterogeneous Information Service Platform

论文作者 _____

指导教师 _____

申请学位 _____ 工程硕士

培养单位 _____ 软件学院

学科专业 _____ 软件工程

研究方向 _____ 项目

答辩委员会主席 _____

评阅人 _____

南开大学研究生院

二〇一三年十月

摘要

本文的研究背景基于航空系统平台。航空系统是一个多学科融合，多子系统集成的庞大的高复杂系统。

新技术的不断成熟，新业务的不断发展，新的需求、新的系统不断嵌入，使得应用程序的适用性和效率大打折扣；系统的集成整合如同挥之不去的阴影，一直困扰着整个航空企业级信息化平台，而尤为严重的是重量级的企业级应用。而本文基于面向服务的体系结构提出了解决这一问题的一系列方案。

本文首先对航空业的企业级应用进行了分析研讨，特别针对其大量遗留异构数据系统和 IT 资产进行了分析和研究，并设计构建异构信息服务平台以解决这个问题。在进行设计工作的同时，本文也逐步深入的探讨了系统解决方案，并进行了建模和实践。

本文基于航空业一个真实的案例，通过一系列基于实际业务的设计原则、模式和技术，阐释了充分利用开放标准，将软件资产展现为服务的结构，并提供了软件资产标准的展现和交互途径。

特别是在异构系统服务平台的整体架构设计方面，本文基于 IBM 成熟的中间件技术，设计实现各个模块细节，最终完成针对航空业的可复用异构信息处理平台。

关键字：异构系统 Web 服务 异构信息服务平台 消息流 面向服务架构

Abstract

This paper base on airline system. As we know, the airline system is multi-disciplinary integrated, multi-subsystem integrated huge systems with high complexity.

As new technology continues to mature, new business continues to develop; new requirement continues to raise, new system continues to implement, all of these make the applicability and efficiency of the application greatly reduce and system integration just like lingering shadow, always plague the entire airline enterprise information platform, especially these complicated enterprise application. And this paper, which is based on Service Oriented Architecture, propose a package of solution to solve this problem.

This paper analyze and research enterprise airline application. Especially for its large number of legacy heterogeneous systems and IT assets, design and build heterogeneous information service platform to solve this problem. During the design work, the paper also do an in-depth exploration of system solutions and carry out modeling and practice.

This paper based on a real case in airline industry. Through a series of practical business-based design principles, patterns and techniques, explain the full use of open standards, the software as a service to show the structure of assets and provide a standard software assets and interactive pathway.

Especially in overall heterogeneous systems service platform's architecture design, this paper based on IBM's proven middleware technology, design and implement the details of each module, finally complete reusable heterogeneous information processing platform for airline industry.

Key Words: Heterogeneous System, Web Service, Heterogeneous Information Platform, Information Flow, Service Oriented Architecture

目 录

第 1 章 绪论.....	1
1.1 研究项目背景	1
1.2 研究项目的目的和意义	2
1.3 实践研究内容	4
1.4 本文组织结构	4
1.5 本章小结	5
第 2 章 核心问题分析.....	6
2.1 SOA 的意义和优点.....	6
2.1.1 SOA 的意义.....	6
2.1.2 SOA 的优点.....	6
2.2 消息中间件	7
2.3 GDS 全球分销系统.....	9
2.4 电子数据交换	10
2.5 本章小结	14
第 3 章 异构信息服务平台整体架构.....	15
3.1 HIP 服务平台概述.....	15
3.2 HIP 服务平台优点.....	16
3.3 HIP 平台架构分析.....	17
3.3.1 服务分销平台	18
3.3.2 Web 服务网关	18
3.3.3 服务中间件	18
3.3.4 消息队列	20
3.3.5 ESB 服务	21

3.3.6 HIP 核心服务.....	21
3.3.7 HIP 功能服务.....	22
3.3.8 HIP 框架服务.....	23
3.3.9 HIP 信息处理服务.....	23
3.3.10 HIP ESB 数据仓库.....	24
3.3.11 HIP 工作数据仓库.....	24
3.3.12 服务注册表和存储库.....	25
3.3.13 安全和目录服务.....	25
3.3.14 系统管理和监控.....	25
3.4 本章小结.....	26
第4章 异构信息服务平台核心实现.....	27
4.1 核心模块实现.....	27
4.2 HIP 核心模块交互流程.....	28
4.3 HIP 框架服务实现.....	30
4.3.1 HIP 异常处理组件.....	30
4.3.2 会话管理模块.....	31
4.3.3 审计服务模块.....	35
4.3.4 性能管理服务模块.....	37
4.3.5 日志管理服务.....	38
4.3.6 异常管理子模块.....	40
4.3.7 参考数据服务.....	41
4.4 HIP 数据服务实现.....	41
4.4.1 电子数据交换解析服务.....	41
4.4.2 工作数据仓库.....	43
4.5 HIP 功能服务.....	44
4.5.1 功能服务组件模型.....	44
4.5.2 在线预定系统.....	51
4.5.3 旅客预订系统.....	53

目录

4.5.4 航班查询系统	54
4.5.5 费用系统	56
4.5.6 票务系统服务	57
4.6 数据流转模式和 HIP 服务平台应用场景	58
4.6.1 从异构系统发送到 HIP 平台的服务请求	58
4.6.2 从 HIP 平台发送到服务分销平台（终端服务提供者）的请求.....	59
4.6.3 从服务分销平台（终端服务提供者）返回 HIP 平台的服务应答.....	59
4.6.4 从 HIP 平台返回异构系统的服务应答	60
4.6.5 由服务分销平台推送到 HIP 平台的旅客定座数据	60
4.6.6 由 HIP 平台服务推送到异构系统的旅客订票信息	60
4.6.7 从服务分销平台向 HIP 平台推送库存清单数据	61
4.6.8 从 HIP 平台服务向异构系统推送库存清单数据	61
4.7 本章小结	62
第 5 章 总结与展望	63
5.1 主要工作	63
5.2 展望	64
参考文献.....	65
致谢.....	67
个人简历 在学期间发表的学术论文与研究成果	68

第 1 章 绪论

1.1 研究项目背景

随着我国航空产业市场化压力的增大，航空产业信息化的需求正日益显现，各大航空公司为提升自身的核心竞争力，都在不断拓展其航空相关业务的信息应用。但是，我国航空产业的信息建设尚处在初级阶段，从应用到网络、从内容到服务都显得很薄弱，公司内部的基础设施建设、电子商务、民航信息化管理等环节信息化水平亟待提升，迫切的需要一整套系统的信息化解决方案。随着市场风云变幻，航空公司为了能在严酷的竞争中生存和发展，信息化建设已迫在眉睫。

但是，航空系统是一个庞大的高复杂系统。其中的经营、管理、市场、销售等各方面都需要广泛应用信息化技术，以提升航空企业的核心竞争力。

在传统的软件工程体系中，需要从需求分析做起，逐步构建各种软件系统，各软件系统有着自己的消息流或者接口，可供业务需求使用。但是这对于多专业集成，多学科融合，多子系统集成的庞大的高复杂性的航空系统工程，并不十分切合。主要原因如下：

随着信息技术的不断革新，需求的变化愈来愈快，产品的周期缩短，市场竞争日趋激烈。而传统软件开发的模式，基于具体的需求，系统化地构建服务。这种模式在面对需求的不断变更和服务的复杂组合时，往往显得力不从心，不能根据市场的瞬息万变，在复杂度很高的集成系统中做出快速的应变。

在另外一方面，航空业已经发展数十载的核心技术和行业标准，以及稳定的应用和系统，势必还会在相当长的一段时期内，参与到信息化建设当中，如何在现有的遗留系统和 IT 资产上构建统一高效的信息服务平台，面向服务的体系结构就是解决这一问题的良方。

1.2 研究项目的目的和意义

图 1.1 是基于消息的传统传递模式的架构，这种消息传递架构的特点是：

- 应用之间点对点的连接。
- 实现简单、基本的信息交互和数据传递。

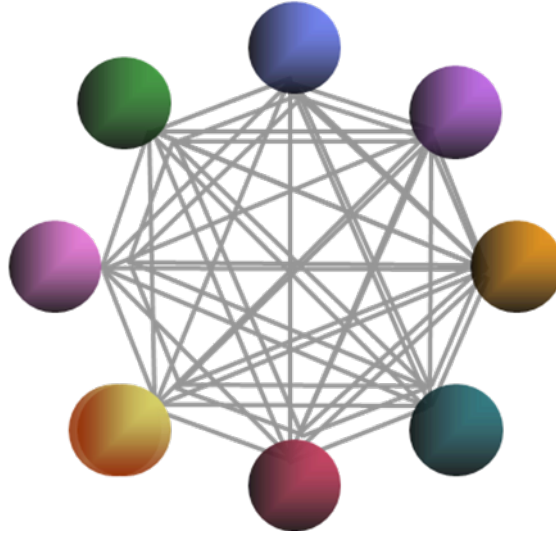


图 1.1 传统消息传递架构

但是，随着系统复杂性的逐步提升和新系统的不断接入，演变出过渡性的消息传递架构，如图 1.2 所示，这种过渡性架构的特点是：

- 通过星形拓扑结构的核心管理节点实现应用之间的整合。
- 很容易管理大量的连接和系统。

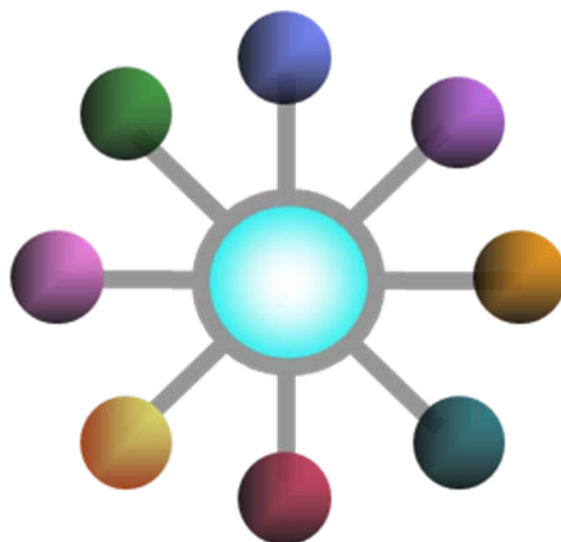


图 1.2 HUB 型消息传递架构

而随着企业业务的快速变化和 demand 增加，信息化的复杂度与日俱增，传统消息框架为每组系统间设置接口和适配器，如果接入系统数量较少，这种点对点的方式复杂性还可以接受，但是接口和适配器与系统的数量成平方关系，后接入的系统必须与每个与自己交互的系统配置接口或适配器，这使得后期的维护和系统的接入复杂度成倍增加。故传统消息传递的架构只适用于较为简单的定制化系统。

相比传统消息传递架构，过渡性框架也需要接口或者适配器，但是他的复杂性与交互系统数量呈线性关系。各系统向中心控制节点获取、推送数据，由中心控制节点管理数据流。这样可以做到有效地扩展和维护，但是也有着灵活性不足，功能与业务紧耦合的缺点。

所以 SOA(service-oriented architecture, 面向服务的体系结构)^[1]的架构模型逐渐在企业级应用开发领域崭露头角。SOA 根据不同的需求，通过网络等方式，对松散耦合的粗粒度应用组件进行分布式的组合、应用和部署。如图 1.3 所示，SOA 消息传递的关键在于“服务”，服务间彼此可以通信，这种通信既可以是简单的数据传送，亦可以是多个服务间协调进行的活动。换句话说，信息资源在 SOA 中被置于标准访问的各种服务中，在消息的消费者看来，这一系列服务的组合就如同消息池，消费者可以按需存取消息，真正地把业务逻辑和底层实现进行了分离。

本文使用日渐成熟的 SOA 构架，针对航空业复杂的信息传递，构建 HIP (Heterogeneous Information Platform, 异构信息服务平台)，通过该平台实现服务的整合集中和流程实现，并且借助标准的接口灵活地连接，实现真正的按需应变。



图 1.3 总线型消息传递架构

1.3 实践研究内容

本文基于面向服务的构架，对航空业的企业级应用进行了分析研讨，特别针对其大量遗留异构数据系统和 IT 资产问题进行了分析和研究，并设计构建异构信息服务平台以解决这个问题。在进行设计工作的同时，本文也逐步深入地探讨了系统解决方案，并进行了建模和实践。

本文基于航空业一个真实的案例，通过一系列基于实际业务的设计原则、模式和技术，阐释了充分利用开放标准，将软件资产展现为服务的结构，并提供了软件资产标准的展现和交互途径。

特别是在异构系统服务平台的整体架构设计方面，本文基于 IBM 成熟的中间件技术，设计实现各个模块细节，最终完成将针对航空业的可复用异构信息处理平台。

1.4 本文组织结构

根据上述研究内容和研究目标，文章其余部分内容组织如下。

第二章：

本章研究了构建异构信息服务平台所需的核心背景和技术。分别阐述了 SOA 的意义和应用；消息中间件的意义和通信方式；航空业独有的全球服务分销系统和它的重要意义，该系统将作为主要的终端服务提供者与异构信息处理平台交互；航空业电子数据交换的主要方式，及其在航空业中的应用。

第三章：

阐述了 HIP 异构信息服务平台的整体架构。

首先是 HIP 平台的概述和其所具有的优势，接下来是各系统各部分模块的详细设计分析。首先是对于服务分销平台的分析；其次是对 HIP 平台 Web 网关服务的设计和实现；然后是针对服务中间件和消息队列企业级产品的选取和分析；最后分别针对 HIP 核心服务、功能服务、框架服务、信息处理服务以及数据仓库进行了较为概括的设计和研讨。

第四章：

详细地阐述了 HIP 核心的各个模块的实现。

本章首先描述了 HIP 平台核心模块的整体架构图；其次阐述了基本消息流中，各核心模块之间的交互过程；然后针对 HIP 框架服务、数据服务和功能服务的设计实现细节，进行了详细的分析和研讨；最后基于平台的实际应用场景，设计实现了数据传输的模式。

第五章：

本章中，我们在总结 HIP 的同时，挖掘不足之处和可以继续改进的地方。

1.5 本章小结

在本章中，我们首先介绍了研究项目的课题来源和项目相关背景，即航空产业的发展分析和面临的主要挑战。由此揭示了异构信息服务平台的重要性以及 SOA 框架的适用性。另外，我们还列出了本文的主要研究内容、研究目标以及文章的组织结构。

第 2 章 核心问题分析

2.1 SOA 的意义和优点

2.1.1 SOA 的意义

首先，本文针对 SOA 的意义和对于复杂系统集成问题的优势进行分析阐述。

SOA 系统平台是将异构平台和系统上应用程序的不同功能部件封装为服务，并通过定义良好的接口和规范把这些服务以松耦合方式整合在一起。简而言之，SOA 旨在使信息系统变得更有弹性，以更快得响应业务单位的需求。

所以，接口应当采取中立的方式进行定义，接口本身独立于实现服务的硬件平台、编程语言和操作系统。这种方式，使得构建在各种异构系统中的服务可以以一种通用和统一的方式进行信息交互^[2]。

这种没有强制绑定到特定实现、特定技术上接口定义的特征，称为服务之间的松耦合。松耦合系统的好处有两点，一点是它的灵活性，另一点是，当组成整个应用程序的服务根据需求变化，本身的内部结构和实现逐渐地发生改变时，复杂服务可以使用基本服务，重新编排组合使用。

针对航空行业的信息平台，SOA 可谓物尽其用。粗粒度的服务接口封装一项具体的业务功能。针对业务的变化（例如促销活动），组合编排复杂服务。针对大量已有的 IT 资产和异构系统，提供松耦合的 Web 服务，以统一标准整合系统，为开发和维护工作节省了大量时间。

2.1.2 SOA 的优点

1. 复用现有的资产。

SOA 可将现有的资产包装成提供功能的服务。组织可以继续从现有的资源中获取价值，而不必重新从头开始构建。

2. 更好的伸缩性和可用性。

服务间的松散耦合关系和服务设计的层次架构模型，可以实现更好的伸缩性和可用性。使用者无需知道提供者的实现具体细节；提供者则可以在企业服

务总线等集群环境中灵活部署。

3. 更易于集成和管理复杂性。

将基础设施、技术更新和具体实现发生的改变所带来的影响降到最低限度，隔离了需求和业务复杂性。当在航空信息平台这种场景下，很多异构系统一起协作提供价值链时，这点就显得尤为重要。

2.2 消息中间件

上一节阐述了 SOA 的意义和优点，本节承上启下，针对 SOA 中最核心的消息中间件进行分析。

消息中间件 (Message Oriented Middleware, MOM) ^[23] 是一种由消息传递机制组成的中间件技术，利用高效可靠的消息传递机制进行与平台无关的数据交流，并基于数据通信来进行分布式系统的集成，具有异步传送、防御通信、并发执行、日志通信、通信方式多样和隔离网络复杂性等多种特性。

目前的消息中间件主要支持两种消息通信模型：消息队列和消息传递^[25]。消息队列是一种间接通信模型，其通信基于队列来完成；消息传递是一种直接通信模型，消息被直接发给感兴趣的实体。这两种消息传递的模型在航空信息平台中，都是不可或缺的。在后面的章节中，HIP 服务平台的设计和实现都会使用这两种消息模型。

在消息中间件中，有两个典型的角色：消息生产者和消息消费者。消息生产者产生和发送消息到消息中间件中，而消息消费者从消息中间件中获取消息进行相应的处理。而对于 HIP 平台，它既是消息的生产者，也是消息的消费者。基于 SOA 理念，当异构系统发出消息请求时，HIP 服务平台将会充当服务的生产者，请求系统无需知道最终消息的生产者是谁；同样当面对消息的生产者时（例如下节阐述的服务分销平台），HIP 平台根据不同的业务分类，存储相关数据，以消费者的角色从上游数据源抽取数据。

这里本文使用到了 Java 消息服务 (Java Message Service, JMS) ^[9]，JMS 是 Java 平台中关于消息中间件的应用程序接口。在 JMS 中，消息由两部分构成，消息头和消息负载。消息头是用来存储与消息相关的元数据的，而消息负载就是消息的实际内容，可以是文本数据、二进制数据，也可以是特定格式的数据。

按照消息在传递过程中是否被序列化，消息的传递模式可以分为持久化型

和非持久化型。持久化型是用于那些对消息传递可靠性要求非常高的情况下，消息在传递过程中被保存到持久化介质（硬盘等）中，在消息中间件发生故障的情况下，消息仍然能够发送且仅发生一次到消息的目的地。非持久化型主要用于可靠性要求不高的消息发送过程中。但是在消息发送吞吐量较高的场景中，由于消息被保存在内存中，当发送出现故障时，消息会丢失，因此消息只能确保最多会被发送一次。

目前的消息中间件的消息传递模式有两种，如图 2.1 和图 2.2 所示，点对点（Point To Point, PTP）模式^[14]和发布/订阅（Publish/Subscribe）模式。点对点的模式中，消息发送的目的地是消息队列（queues）。消息队列中接受到的消息，只会被传递给一个并且是唯一的消息消费者，非常类似于个人通过电子邮件服务器发送人对人的电子邮件。而发布/订阅模式的消息发送目的地是主题（topics）。消息生产者将消息发送到某一主题中，消息消费者通过注册订阅该消息来接受该主题中的消息。该模式是典型的一对多的关系，任何被发送到某主题中的消息都会被传递给所有的订阅者，类似于发电子邮件给邮件列表中的所有用户。

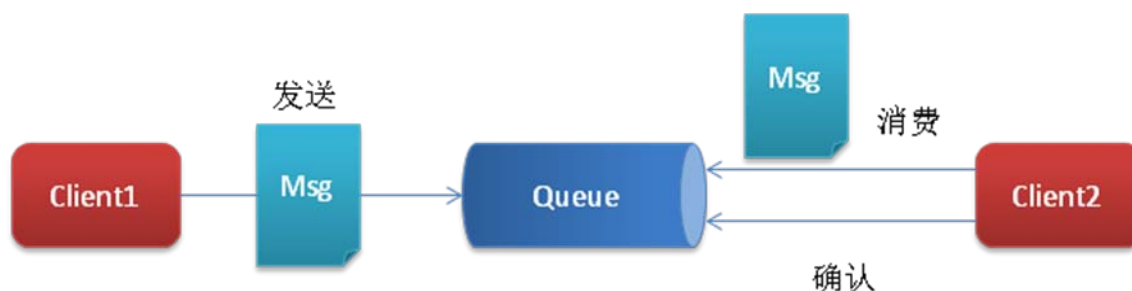


图 2.1 PTP 点对点传输

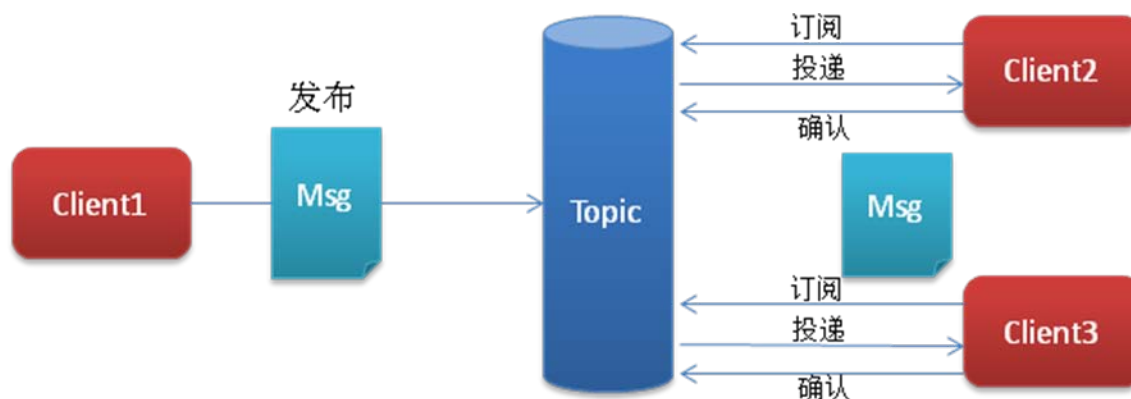


图 2.2 Publish/Subscribe 发布订阅模式

2.3 GDS 全球分销系统

GDS (Global Distribution System, 全球分销系统), 是应用于航空行业及旅游行业的大型计算机信息服务系统。通过全球分销系统, 全球的旅游销售机构可以及时地从航空公司、旅馆、租车公司、旅游公司获取大量的与旅游相关的信息, 从而为顾客提供便利、快捷、可靠的服务。航空产业主要使用的分销系统有 AMADEUS、ABACUS、GALILEO、SABRE, 在我国航空产业, 主要使用中国航信的分销系统 TRAVELSKY。

从 GDS 的发展过程看, GDS 是由于旅游业的迅猛发展而从航空订座系统中分流出来的面向旅行服务的系统。如今, GDS 已经发展成为服务于整个旅游业的一个产业, 除了原有的航空运输业, 旅馆、租车、旅游公司、铁路公司等也纷纷加入到 GDS 中来。经过技术与商务的不断发展, GDS 已经能够为旅行者提供及时、准确、全面的信息服务, 并且可以满足消费者旅行中包括交通、住宿、娱乐、支付及其他后继服务的全方位需求^[35]。

GDS 作为整个航空系统销售的数据源, 其分发的数据会被多个下游系统使用, 可以支持复杂多变的业务组合。所以 GDS 系统是 HIP 服务平台的上游, 他的主要数据分发方式将在下一节进行讨论。而与整个平台的整合设计将在第三章进行阐述。

2.4 电子数据交换

航空业的数据交换格式主要采用电子数据交换格式，所以本节针对这种特定的格式进行分析阐述。

电子数据交换 (Electronic Data Interchange, EDI)^[11]，国际标准化组织 (International Organization for Standardization, ISO) 对它的定义如下：将商业或者行政事务，按照一个公认的标准，形成信息数据结构或结构化的事务处理，从计算机到计算机的电子数据传输。而比较简单的说法是通过电子方式，将组织间的数据进行结构化传递，从而被用来从一个计算机系统到另一个计算机系统传递电子文档或者业务数据，而不需要人工的介入。在贸易领域，EDI 有着更加形象直观的称呼：“无纸贸易”，从中可以看出 EDI 的电子化传递的数据与传统的纸质贸易之间的共性和区别。

EDI 报文是一种结构化的数据格式，在航空行业被广泛的应用，主要有两种标准：

- 美国国家标准化委员会 (American National Standards Institute, ANSI) 推出的 ANSIX. 12。
- 联合国组织并颁布的 EDIFACT 标准 (Electronic Data Interchange For Administration, Commerce and Transport)。

在航空行业的服务分销平台，主要使用 EDIFACT 标准进行信息的传递。

图 2.3 和图 2.4 是一个采购订单的纸质单据和其对应的 EDI850 报文。

第 2 章 核心问题分析

Purchase Order Number: PONUM-1234567				
Sample Ship to Customer			Customer No.	
Ship to: 1010 Any Street Anycity, IL 12345			999999	
Sample Bill to Customer				
Bill to: 2030 Another Street Anycity, IL 12345				
Delivery notice: DO NOT SHIP BEFORE 02/05/98				
Round up to next carton quantity				
ISBN	Author Title	Year	Qty	Price/Disc.
0835209881	Sample Author Sample Title of the Book	97	12	29.00/43%
0835210022	Another Author Title of Another Book	98	23	29.50/43%
0835210456	Last Author Title of the Third Book	98	32	26.00/43%
0835210979	Sample Author Title of the Fourth Book	97	34	24.00/43%
0835211614	Another Author Title of the Fifth Book	98	20	6.99/43%
0835212653	Last Author Title of the Sixth Book	98	40	5.50/43%
0835209784	Sample Author Title of the Seventh Book	97	34	6.99/43%
0835213153	Another Author Title of the Last Book of this Order	98	18	8.95/43%
			Total Items	213
			Total Retail Value	\$3,433.06
			Total Value of PO	\$1,956.84

图 2.3 采购订单纸质版本实例

```

ST*850*0001~
BEG*00*NE*PONUM-1234567**980131**AE~
CSH*B~
DTM*037*980205***19~
SI*BI*CR*BI~
N1*ST**15*6678157~
N1*BT**15*1135309~
REF*PD*PROMO-987~
PO1*1*12*UN*29*SR*IB*0835209881~
CTP*****DIS*.57~
PO1*2*23*UN*29.5*SR*IB*0835210022~
CTP*****DIS*.57~
PO1*3*32*UN*26*SR*IB*0835210456~
CTP*****DIS*.57~
PO1*4*34*UN*24*SR*IB*0835210979~
CTP*****DIS*.57~
PO1*5*20*UN*6.99*SR*IB*0835211614~
CTP*****DIS*.57~
PO1*6*40*UN*5.5*SR*IB*0835212653~
CTP*****DIS*.57~
PO1*7*34*UN*6.99*SR*IB*0835209784~
CTP*****DIS*.57~
PO1*8*18*UN*8.95*SR*IB*0835213153~
CTP*****DIS*.57~
CTI*8*213~
SE*26*0001~
    
```

图 2.4 采购订单 EDI 报文实例

在 EDI 报文中，数据是一串有序的字符序列，是通过一系列的数据段（Segment）和数据段终止符（在上面的例子中是符号“~”）组成的。例如，“ST*850*0001~”指的就是报文的传输集头，而数据段都是由一系列的基本数据元素（Element）组成，数据元之间通过字符“*”进行分割，其中 ST 用于标明开始传输 EDI 报文，后面的 850 表示传输的是一个采购订单，之后的 0001 是相应的唯一控制码。简单地说，EDI 报文的基本结构是首先由数据元和数据元分隔符构成数据段，之后由数据段和数据段终止符构成完整的报文序列。

而 EDI 报文中数据段和数据元的含义是通过数据传输集规范定义的，如图 2.5 所示。

BEG

Beginning Segment for
Purchase Order

Pos: 020 Max: 1
 Heading - Mandatory
 Loop: N/A Elements: 4

User Option (Usage): Must use
Purpose: To indicate the beginning of the Purchase Order Transaction Set and transmit identifying numbers and dates

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max	Usage
BEG01	353	Transaction Set Purpose Code	M	ID	2/2	Not used
<i>Description:</i> Code identifying purpose of transaction set						
Code Name						
00 Original						
BEG02	92	Purchase Order Type Code	M	ID	2/2	Must use
<i>Description:</i> Code specifying the type of Purchase Order						
Record Name: Header Position 187						
Example: po_type						
User Note 1: <i>If your company sends codes other than those specified, please consult the EDI contact person.</i>						
Code Name						
BE Blanket Order/Estimated Quantities (Not firm Commitment)						
DS Dropship						
User Note 1: <i>Seagate will support Drop Ships and the use of Drop Ship code 'DS' at a future date. If Purchase Order product is intended to be drop shipped and 'DS' code is included, then N103 and N104 should not contain values.</i>						
NE New Order						
RL Release or Delivery Order						
SA Stand-alone Order						
BEG03	324	Purchase Order Number	M	AN	1/22	Must use
<i>Description:</i> Identifying number for Purchase Order assigned by the orderer/purchaser						
Record Name: Header Position 129						
Example: po_number						
BEG05	373	Date	M	DT	8/8	Must use
<i>Description:</i> Date expressed as CCYYMMDD						
Record Name: Header Position 179						
Example: po_date (MMDDYYYY)						

Semantics:

- BEG05 is the date assigned by the purchaser to purchase order.

User Note 1:
*Sample Data: BEG*00*NE*CH2372853**100105~*

图 2.5 传数据规范片段

规范中对 EDI 报文中出现的数据段进行了解释，对数据段中各种位置上的数据元的名称、是否必填、数据类型、字符个数的限制等都进行了详细的定义

和说明。

EDI 报文不仅仅是一个数据的标准，更是报文发送者和报文接受者之间指定的协议的具体体现，还有相应的软硬件和传输网络，是一整套的规范和协议^[5]。

图 2.6 是一个典型的两个企业间的通过 EDI 进行贸易的过程。

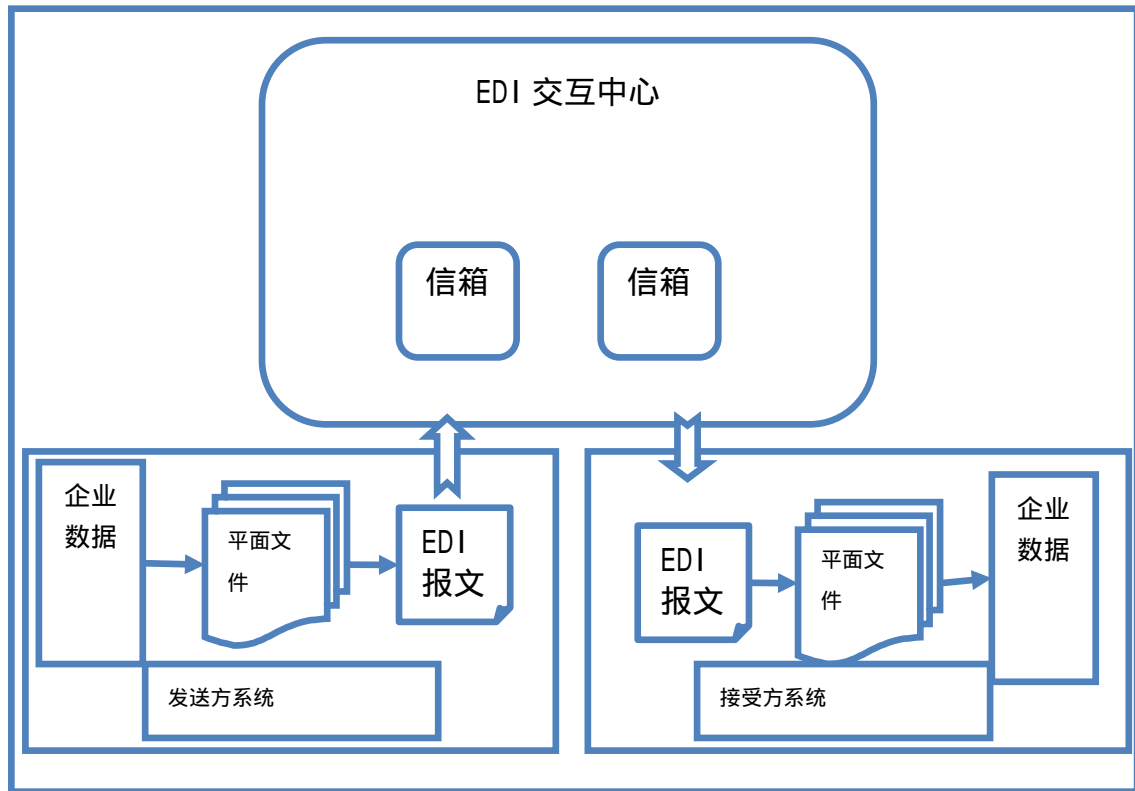


图 2.6 传统 EDI 工作流程

传统的 EDI 系统分为用户系统和中心系统两部分。发送方用户系统主要负责将数据从企业信息化系统中提出来并生成平面文件，然后通过格式转换将平面文件翻译成标准格式的报文，通过发送方用户系统的通讯模块将标准格式的报文发送到 EDI 交互中心系统。而 EDI 交互中心系统通过存储转发的机制将报文推送到接受方系统，或者是接受方系统主动从 EDI 交互中心提取自己的报文，再将标准报文翻译成平面文件，通过映射将平面文件转换成接受方企业能识别的数据。

通过对 EDI 的报文结构和其工作原理上的分析，EDI 报文并不是一种强大的异构数据源之间的交互手段，其价值的产生根源是在不同组织之间经过协商、洽谈和协调上产生的协议规范之上的，所以 EDI 的使用需要 EDI 报文发送和接

受双方都遵守相关的协议。EDI 中蕴含的公共传输标准和协议是处理当今复杂多变的异构数据源的一个重要解决思路。

由于航空平台已有系统对于 EDIFACT 形式信息的广泛应用, EDI 的使用和解析成为 HIP 服务平台重要的一个模块。另一方面, HIP 服务平台可以有效屏蔽不同异构系统不同协议规范对于数据传输影响, 以弥补 EDI 信息在异构数据交互上的短板, 有效地复用已有的 IT 资产。

2.5 本章小结

本章研究了构建 HIP 所需的核心背景和技术。分别阐述了 SOA 的意义和优点; 消息中间件的意义和通信方式; 航空业独有的全球服务分销系统和它的重要意义, 该系统将作为主要的终端服务提供者与异构信息处理平台交互。最后分析了电子数据交换的主要方式、核心技术和背景, 结合航空行业实际应用进行了讨论, 为整个 HIP 平台的实现奠定了基础。

第 3 章 异构信息服务平台整体架构

3.1 HIP 服务平台概述

HIP 平台是针对航空预定系统的集成平台，是产品、组件、服务和工具的整合平台，适用于各种复杂航空行为，并且可以无缝的与各种服务分销平台进行信息交互。

HIP 集成平台有着以下几个关键特征：

1. 包含富应用服务集合，应用服务不但包括基本针对特定业务系统、特定数据类型的增删改查，还包括经过编排组合过的复杂服务。HIP 集成平台可以有效地集成各种不同的服务系统，不同服务分别为预定系统，离港控制系统，库存清单管理系统等而设计。

2. 包含框架服务集合，可以提供集成的内部服务，以支持企业级应用维护、监控和管理。

3. 包含大量消息流，智能地转化多种异构信息类型，建立终端服务提供者和异构服务消费者之间的桥梁。另外，HIP 可定制客户的配置，优化路由，转化消息流以及中介流。

4. 把功能性服务发布为标准的 Web 服务，可根据不同情况，使用 SOAP/HTTPS 进行访问。并且提供跨越多个域的服务访问（Session 实现）。

5. 使用了 IBM 的 Websphere Message Broker 消息中间件和 Websphere MQ 产品，并针对 HIP 集成平台进行了定制化的开发。

6. 针对航空数据的特定标准（Edifact），提供信息分析模块，可以解析、验证并帮助导入 edifact 数据到数据模型仓库。

7. 包含数据模型仓库，以存储相应的逻辑模型和业务相关数据（乘客信息数据、库存清单数据、离港信息数据等）。

8. 工作数据仓库模型包括

- PNR（旅客订座记录）
- 库存清单管理
- 离港信息 - 客户信息

- 离港信息 - 座位信息
 - 离港信息 - 行李信息
 - 离港信息 - 其他信息
- 等等航空系统相关的业务数据。

3.2 HIP 服务平台优点

与航空业目前的一些系统相比，HIP 系统集成平台，主要有以下几个优点：HIP 服务平台基于一个成功的商业案例，系统和设计方案已经过测试，并已经上线。

HIP 服务平台可灵活配置。审计、日志、性能管理、服务超时等核心因素，都可以进行简单直接的配置来满足客户的需求。

HIP 服务平台提供了大量标准的功能性服务，这些服务都是基于基本的航空业务需求，并且具有一定的代表性。各类服务发起的异构系统，可以直接调用功能性服务消息流，通过配置正确的服务终端提供者，获取所需信息。与异构系统和服务提供者的无缝连接将使用下列功能性服务。

- 传入消息解析。
- 消息的验证。
- 调用服务提供者的多种方式。
- 动态路由到实际地址。
- 日志、审计、异常和错误管理等。

以及对于安全目录服务、服务注册表和存储库等外部系统的整合等。

HIP 服务平台基于成熟的 ESB 产品构建-IBM - Websphere Message Bus 和 Message Queue。提供一个稳定、可扩展和可靠的平台，以便于支持高容量、事务性的实时应用程序。

HIP 服务平台的另一个创新点在于特别的会话管理模块。根据服务消费者不同的业务需求，不同的业务逻辑和不同的权限和组别，会话控制提供不同的会话类型和会话管理。针对长会话，会话管理进行支持并优化处理。

HIP 服务平台的消息解析模块提供了通用的框架以及可扩展的应用程序，提升了解析和导入特定格式（EDIFACT）信息的速度，并根据需求，提供实时解析处理/批量解析处理两种不同处理方式。

HIP 服务平台有着完善的错误和异常模块、监控模块以及性能模块；在企业级的系统分析运维中，这几个模块起到了重要作用。

HIP 服务平台提供了工作数据仓库模型，以存储服务分销平台发送的不同数据。该工作数据仓库模型包括逻辑设计模型和物理设计模型，可以使得各不同格式的消息模块和消息内容更好的进行映射和转化。

3.3 HIP 平台架构分析

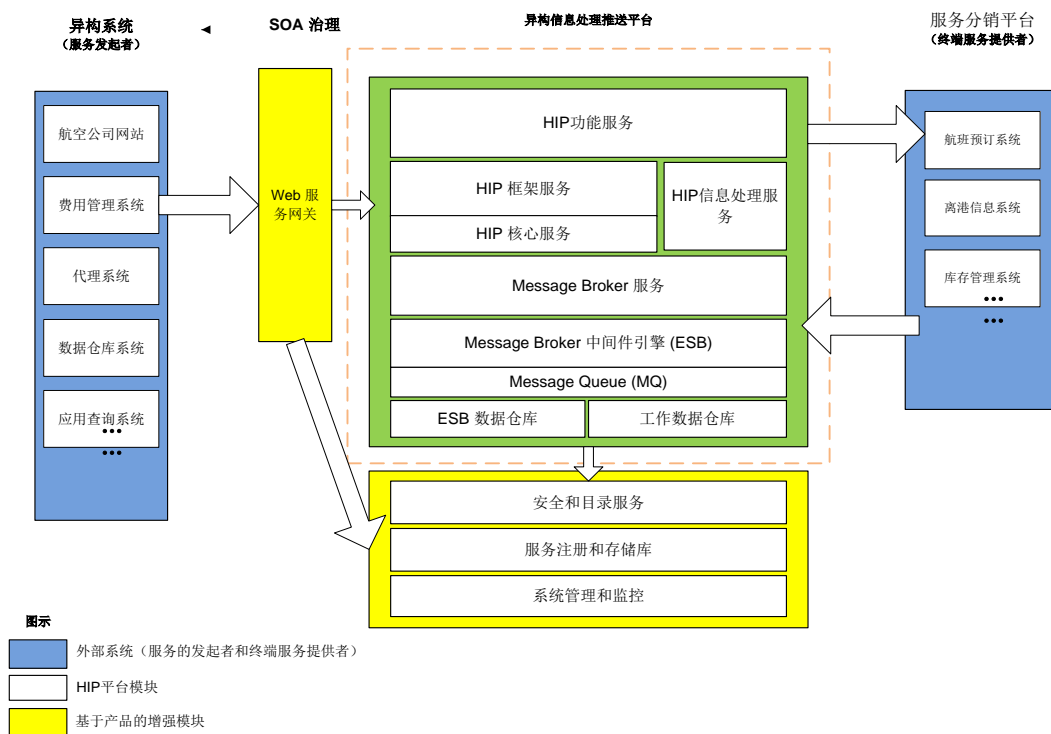


图 3.1 HIP（异构信息服务平台）平台架构

本节分析阐述 HIP 服务平台的整体架构。

如图 3.1 所示，整个架构分为三个系统组：HIP 服务平台组、基于产品的增强模块以及各种外部系统和服务分销平台组。

异构系统即服务发起者，也就是各种服务的消费者。异构系统将向 SOA 治理层的 Web 服务网关发送信息，经过验证-授权-审核，消息请求被发送到 HIP 异构信息处理平台。

HIP 平台包括业务处理使用的 HIP 功能服务集合，支持平台各项功能的 HIP 框架服务和核心服务功能，支持 EDI 消息处理的信息处理服务，经过设计开发的中间件服务以及使用的消息队列，另外，还包含服务总线的数据仓库以及义务相关的数据仓库。

除此之外，HIP 服务平台基于 SOA 的理念，扩展提供了安全和目录服务，服务注册和存储库，系统管理和监控等模块。

HIP 服务平台将从服务分销平台的不同系统中，采用多种方式（实时、批量等）抽取数据。

下面就各个模块进行分析概述。

3.3.1 服务分销平台

服务分销平台是商业服务的提供者。在本文中主要体现为航空预订系统，也被称作旅客支持系统（Passenger Support System, PSS），主要含有三个模块：1. 预定系统 2. 离港控制系统 3. 库存清单管理系统。在这里，服务分销平台就是旅客支持系统的一个例子，特征化地支持着三个服务模块。所以服务分销平台是终端的服务提供者，当 HIP 获取任意需求或查询后，将转化并发送请求到服务分销平台，服务分销平台将返回相应的数据到 HIP，并由 HIP 推送服务到相应的异构系统。

3.3.2 Web 服务网关

Web 服务网关^[3]，也被称作安全网关和虚拟服务，作为 HIP 集成平台所有 Web 服务请求的一个登入节点，提供 AAA（验证 - 授权 - 审核）框架（Authentication-Authorization-Audit）以及服务路由的功能。从异构系统发出的服务，首先送入 Web 服务网关进行验证和路由，路由至相应的 Web 服务进行处理。服务网关可通过多种方式实现验证和授权，并支持 Broker-based（基于经纪人方式，这种技术的特点就是，有一个集中的认证和用户帐号管理的服务器）和 Agent-based（基于代理人方式）的单点登录。

3.3.3 服务中间件

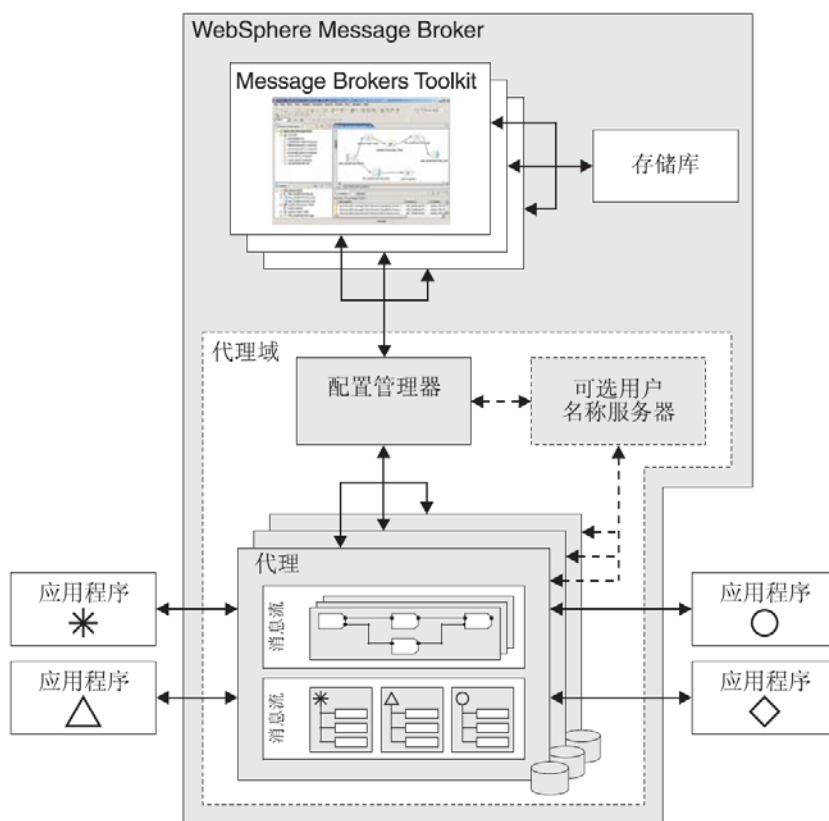


图 3.2 WebSphere Message Broker 消息中间件架构

服务中间件模块，是 ESB（Enterprise Server Bus，企业服务总线）的相应实现^[4]。

在这里，HIP 服务平台通过服务中间件发布 Web 服务，动态地进行路由，进行数据转换和推送。

HIP 服务平台使用企业级 IT 解决方案佼佼者 IBM 的服务中间件，即 WebSphere Message Broker，作为消息中间件引擎，WebSphere Message Broker 是一款企业服务总线（ESB）产品，为面向服务架构（SOA）环境甚至非 SOA 环境提供连通性和通用数据转换。可以支持跨平台、跨硬件的不同应用，从业务逻辑到数据流的转化和映射。如图 3.2 所示，业务逻辑和数据推送接受的逻辑，以及两者之间的映射，都可以在 message broker 中通过转化、路由和中介流实现。该消息中间件提供 SOA 框架下异构应用和服务之间的联系整合。

WebSphere Message Broker 包含以下模块：

- WebSphere Message Broker runtime。
- WebSphere Message Broker Toolkit。

- WebSphere Message Broker Explorer。

HIP 服务平台利用 IBM Message Broker 可以实现下列目标。

1. 利用 Message Broker 健壮的功能来解决各种集成需求，从而满足任何规模项目的需要。
2. 在通过数据流经业务应用和系统时，利用 Message Broker 的开发工具，提供快速数据访问、可视化操作和控制，帮助整个组织制定更明智的业务决策。
3. 连接各种异构应用和 Web 服务，无需进行复杂的点对点连接。
4. 为实现标准化的、简单而灵活的集成提供基础，更加快速和轻松地响应业务需求，并能够随业务的发展而扩展。

为了有效利用消息中间件引擎，HIP 平台将做以下优化：

- ✓ 对流入消息，路由分发到多个目的地。
- ✓ 支持多种可选的消息转化格式。
- ✓ 实现消息的分发聚合。对于复杂请求消息，拆解消息并分别发送到目的地，然后根据不同的回复，拼接回复内容，再返回给复杂请求消息的调用者。
- ✓ 调用 Web 服务获取数据。
- ✓ 多粒度的事件分析以及错误处理。
- ✓ 使用订阅/发布模型提供内容和主题的消息路由。

3.3.4 消息队列

HIP 服务平台和 Message Broker 消息中间件使用 WebSphere MQ 作为消息引擎。

IBM MQ (IBM Message Queue) 是 IBM 的一款商业消息中间产品，适用于分布式计算环境或异构系统之中^[4]。消息队列技术是分布式应用间交换信息的一种技术。消息队列可驻留在磁盘或者内存中，消息队列存储传输的消息内容，直到它们被应用程序抽取。通过消息队列，应用程序可异步独立地执行——它们不需要知道彼此的关系，在继续执行前也不需要等待接收程序接收此消息的反馈。

消息队列 (MQ) 是一种应用程序对应用程序的通信方法。应用程序通过写和检索出入列队的针对应用程序的数据 (消息) 来通信，而无需专用连接来链

接它们。消息传递指的是程序之间通过在消息中发送数据进行通信，而不是通过直接调用彼此来通信，直接调用通常是用于诸如远程过程调用的技术。排队指的是应用程序通过队列来通信。队列的使用除去了接收和发送应用程序同时执行的要求。

IBM WebSphere Message Queue 产品支持应用程序通过不同组件，如子系统、处理器、操作系统以及通信协议的网络彼此进行通信。

3.3.5 ESB 服务

HIP 平台基于 Websphere Message Broker 实现的服务如下。

路由：提供基于消息内容、系统调用服务方式等核心因素的智能路由。

信息转换：把从服务消费者得到的消息结构和内容转变为服务提供者的理解的模式。信息转换是非常重要的一个功能，尤其是在多个系统和应用没有统一的接口时。

信息推送：ESB 基本的信息传输模块，用于各组件传输数据。信息推送由类似 WebSphere MQ 的消息中间件应用提供。消息推送为异步过程，在此基础上中间件技术提供了额外的优势，例如队列、验证以及消息的一次传输。

服务编排：HIP 平台提供服务的编排，可以向终端服务提供者并行或顺序发送多个服务请求，再进行聚合。

3.3.6 HIP 核心服务

核心服务是一系列内部服务，被设计实现为平台其他模块提供基础服务，并且构建平台基本的 ESB 内部架构。所以核心服务模块是标准化并且可重用的。

环境变量：

消息中间件包含一系列环境变量，需要根据不同情况进行配置。环境变量包括调节参数、时间阈值、外部引用 URL 等。在消息流模块配置中，对环境变量进行配置以实现松耦合和高可维护性。

通用参考&映射数据：

提供了一个容器存储系统范围的参考数据，并且提供查询服务查询数据。不同接口间可能需要特定的映射规则，关联映射表定义了数据映射时的参考。

关键映射：

定义了一个跨越多个系统来映射键值的机制，以便正确的为每个特定接口存储键值和查询键值。

框架组件：

顺序或者并行编制其他框架组件（比如异常处理、审核、日志记录、消息验证、报告报表以及会话管理模块）。

3.3.7 HIP 功能服务

HIP 的功能服务与其他核心服务直接相关。HIP 功能服务被其他异构系统调用，从终端服务提供者处获取信息，聚合组织返回调用者。这个过程成为一个商业行为或用例。在航空行业，HIP 功能服务的例子为：建立旅客乘机信息，得到电子客票信息等行为。根据实际需求，HIP 应为 24 全天候服务。

HIP 发布的预约服务是一系列可复用的业务服务，有着以下特征。

- 被各类异构业务信息和外围系统调用。
- 在服务提供者和服务消费者之间提供中间服务，以实现服务的松耦合。
- 服务的提供者等重要属性可以通过参数配置直接更改，具有灵活性；对于生产环境和测试环境的切换，服务负载平衡，以及容灾都有着重要意义。
- 服务即可以被旧有的和遗留的预约系统调用，也可以直接被新的预约平台调用。
- 发布为基于 Http 协议的 Web 服务，在 IBM WebSphere Message Broker 上进行整合开发。

具体服务可分为以下几个子类：

- 在线预订
- 旅客预订服务
- 航班服务
- 费用服务
- 票务服务
-

3.3.8 HIP 框架服务

HIP 集成平台利用底层的 WebSphere Message Broker 和 MQ 基础技术提供了框架级别的技术服务和功能，这些服务大致分类如下。

- 核心服务：这些都是通用组件，负责 HIP 平台中存储环境变量、数据映射信息、关键值对和其他服务编排信息。
- 审计服务：在数据库记录服务请求者和提供者之间的请求响应。
- 日志服务：为所有组件提供标准化的日志记录功能。
- 异常处理服务：提供集中的异常管理组件，为失败的消息提供日志机制，以进行分析处理。
- 会话管理服务：根据客户端的需求，能够创建不同类型的会话。例如联系上下文的会话、专用的会话等。
- 参考数据服务：管理其他服务需要的参考数据。例如 HIP 应用的配置数据，消息流的控制结构等。
- 服务编制服务：HIP 提供服务编制功能，可以顺序或者并行地调用服务提供者以获取信息，再在消息流中进行服务返回结果的聚合。
- 性能测量服务：性能统计数据主要从 ESB 收集记录。
- 性能数据：系统级别的性能指标，可用于生成统计数据和报告。

3.3.9 HIP 信息处理服务

信息处理模块提供了从服务分销平台解析、验证和导入 EDIFACT 模式数据的通用框架服务。

当前，所有主要航空公司都使用旅客预订平台（PSS）管理库存清单、预订服务以及离港控制。而 EDIFACT 是一个标准的交换文件格式，大量用于终端服务提供者向服务消费者传输信息。

消息处理组件目前提供一个基于 EDIFACT 的中间层解决方案，避免由于多个异构信息系统的多次调用，重复处理相同的 EDIFACT 信息。在 HIP 平台中，EDIFACT 信息的解析和加载将存储到工作数据仓库，以响应不同系统的数据请求。处理信息主要有两种方式。

- 实时处理：根据业务需求，某些 EDIFACT 数据（例如乘客值机信息）需要实时地进行处理。实时处理服务对消息进行实时解析，并在关系数据库进行数据插入和更新。
- 批处理：使用批处理服务，消息可以成批的进行处理，导出为平面文件（例如 CSV 格式），然后使用特定的数据库载入数据服务（例如 SQLLoader）导入数据库。

HIP 信息处理服务的优点在于有效地减少异构服务请求系统请求的数量，进而减少开支；并且可以发布为 Web 服务，进行复杂的数据转化聚合。

3.3.10 HIP ESB 数据仓库

HIP 集成平台基于 ESB 构建，维护 ESB 数据仓库以存储中间件需要的配置信息和技术参数。数据仓库主要包含以下几类内容。

- 参考数据：例如各系统需要的配置数据，映射数据代码，业务逻辑规则参考等。
- 审计数据：ESB 处理请求和响应过程中的审计数据。
- 异常数据：由异常处理模块生成的异常数据，这里的数据存储是为了控制消息重放规则和获得错误统计信息。
- 性能数据：存储性能相关数据。

注意：业务逻辑和应用数据不在这里存储，业务和工作数据（例如 EDIFACT 信息数据）在工作数据仓库存储。

3.3.11 HIP 工作数据仓库

HIP 集成平台为存储业务和应用数据，维护了关系数据库和数据仓库，在这里我们统称为工作数据仓库。工作数据仓库存储了从服务分销平台获取的各种模式信息（例如 EDIFACT）。工作数据仓库可以被 EDIFACT 信息处理模块直接调用，以存储同步业务信息。工作数据仓库有着以下特征：

- 从异构系统和终端服务提供者获得的不同格式（EDIFACT、XML、CSV 等）数据中，生成并维护逻辑数据模型。
- 模型包括不同航线库存清单系统、旅客乘机系统以及离港信息系统的逻辑结构。

- 模型基于系统和业务逻辑建立。
- 基于真实的服务分销平台（Amadeus）建立消息规范。

3.3.12 服务注册表和存储库

服务注册表和存储库提供了 SOA 服务的管理和治理。

服务注册表和存储库是用于集中存储服务元数据的服务注册中心和数据仓库。这里的元数据既包括企业 IT 层面的数据，如 WSDL、XML Schema、WS-Policy 或者 SCA（Service Component Architecture，服务组件框架）文件等，也包括企业的业务层面的数据，如流程、服务、契约等。当服务元数据注册在 WSRR（WebSphere Service Registry and Repository，WebSphere 服务注册表和存储库）中后，用户就可对服务元数据进行可视化的操作、版本的管理及使用监控。

3.3.13 安全和目录服务

安全和目录服务管理 Web 服务网关模块递交的所有应用程序的安全请求，目录服务基于 LDAP（Lightweight Directory Access Protocol，轻量级目录访问协议）服务，LDAP 目录中可以存储各种类型的数据，如电子邮件地址、路由信息、库存资源数据、公用密钥、联系人列表等。通过把 LDAP 目录设为系统集成中的一个重要节点，可以简化 HIP 集成平台内部查询信息的过程。

3.3.14 系统管理和监控

系统管理和监控模块以端对端的方式，提供系统健康的监控和管理服务。本模块对于应用程序的实时运行和负载均衡以及服务总线的运行也会进行监控管理。

3.4 本章小结

本节阐述了 HIP 的整体架构。

首先是描述了 HIP 服务平台的概况，对 HIP 平台所具有的优势进行了分析。接下来是各系统各部分模块的详细设计分析，其中首先是对于服务分销平台的分析；其次是对于 HIP 平台 Web 网关服务的设计和实现以及对于 HIP 平台内部功能的概述；然后是针对服务中间件和消息队列企业级产品的选取和分析；最后分别针对 HIP 核心服务、功能服务、框架服务、信息处理服务以及数据仓库进行了较为概括的设计和研讨。

第 4 章 异构信息服务平台核心实现

4.1 核心模块实现

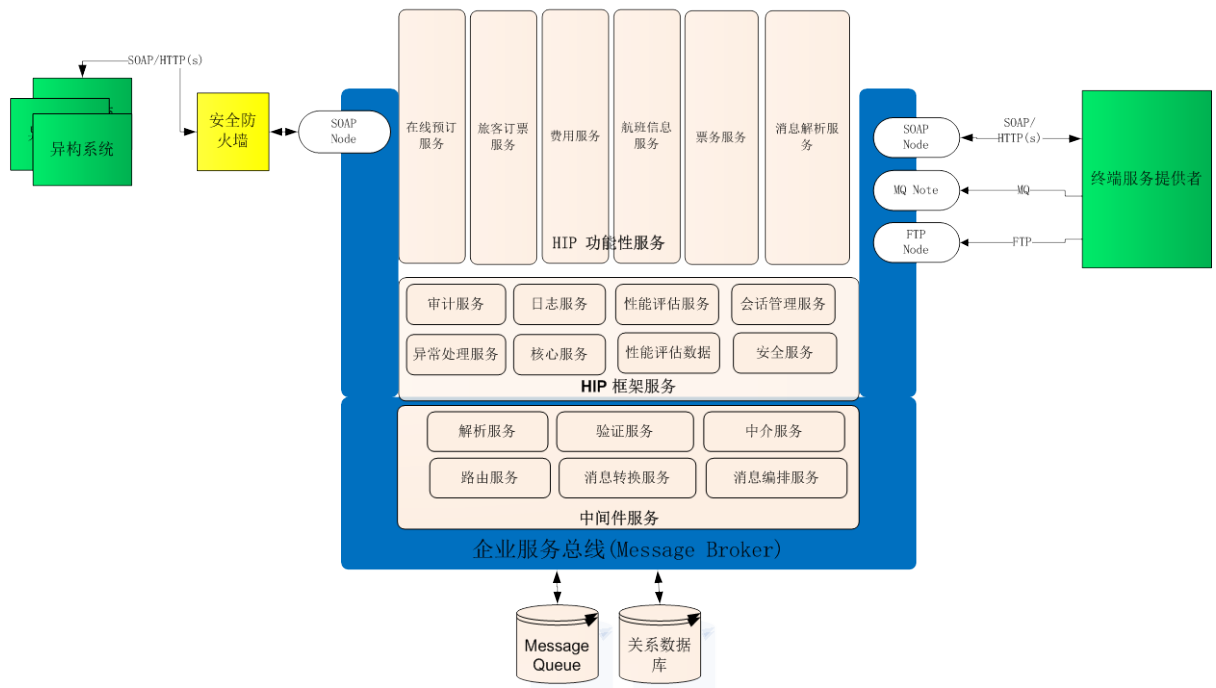


图 4.1 HIP（异构信息服务平台）核心架构

HIP 核心模块整体架构实现如图 4.1 所示，下面将分别阐述为了实现一个基本的消息流，HIP 核心模块交互的流程；支持整个 HIP 服务平台的各种框架服务实现；HIP 服务平台核心的数据服务实现；基于实际需求的业务服务实现，以及 HIP 服务平台支持的信息传递场景和数据流转模式。

4.2 HIP 核心模块交互流程

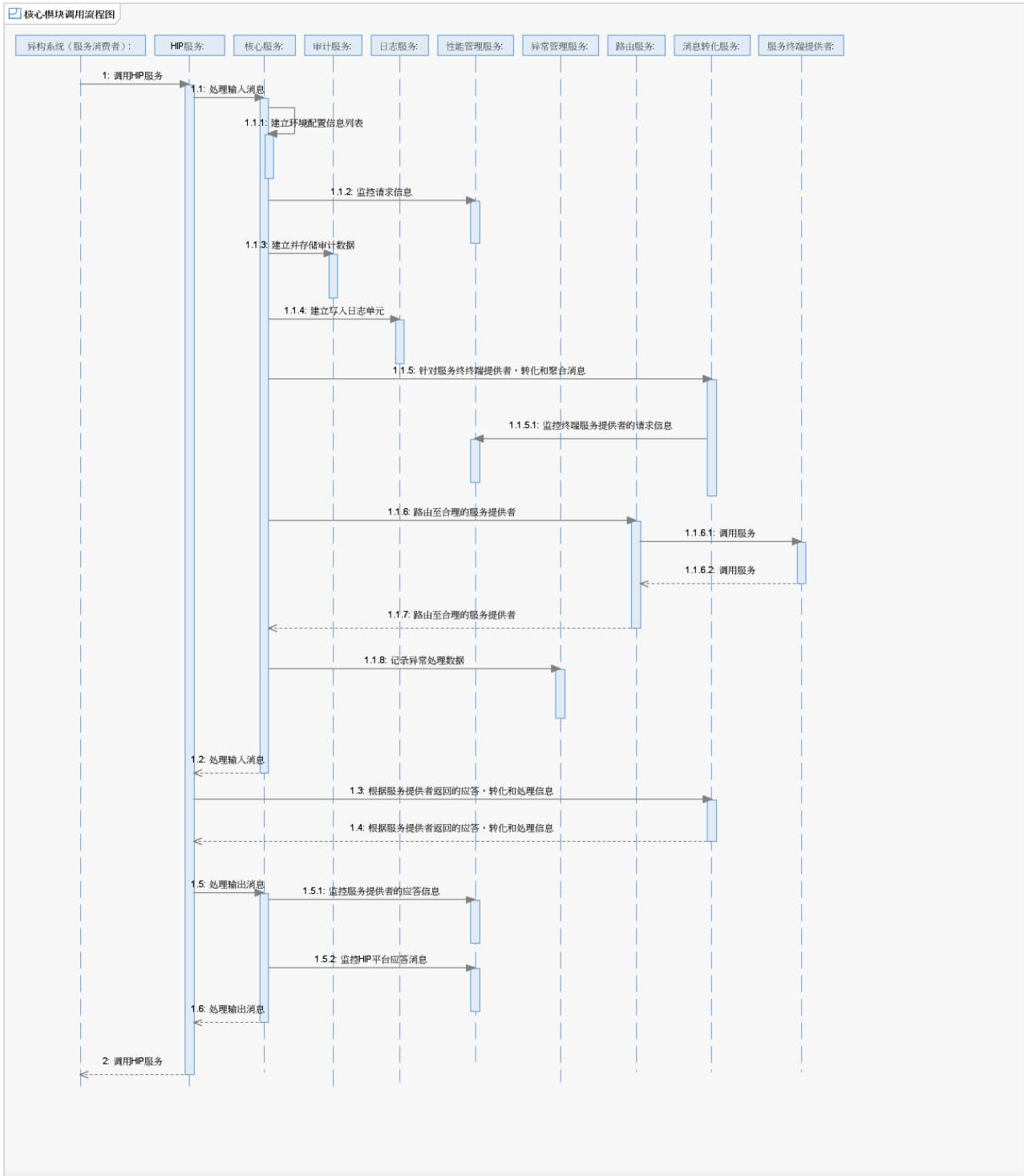


图 4.2 HIP（异构信息服务平台）核心模块交互流程

如图 4.2 所示，HIP 核心模块交互过程的详细实现如下，这里 HIP 的核心模块含有 HIP 的业务服务模块，核心服务模块，框架服务模块（包括审计服务模块、日志服务模块、异常处理服务模块、性能管理服务模块），路由模块以及消息解析转化服务模块，所有的这些核心模块是构建一个完整的消息流所必需的。该核心模块交互流程描述了从异构系统发出请求开始到异构系统接收到实时的请求应答的一个基本的消息流。

1. 异构系统（服务消费者）使用基于 HTTP（HTTP-Hypertext transfer protocol，超文本传输协议）或者 HTTPS（Hypertext Transfer Protocol over Secure Socket Layer，安全的超文本传输协议）的报文请求引入 HIP 功能服务。
2. HIP 的核心服务根据具体需求，初始化环境变量。
3. 接下来性能监控管理模块被调用，分析记录请求开始时间以及请求内容，并进行性能静态分析。
4. 审计模块接着被调用，审计并存储请求信息到数据库。
5. 日志模块被调用，为维护和调试工作生成调用记录。
6. 接下来是消息转换服务，该服务把异构系统的消息模式转化为服务提供者的消息模式，同时，一些复杂义务逻辑的编排组合也在这步完成。
7. 智能路由服务将根据需求，确定路由服务地址。
8. 性能监控模块被再次调用，以记录处理后的请求信息发给终端服务提供者的时间点。
9. HIP 平台把请求信息发给终端服务提供者。
10. 服务提供者返回应答信息给 HIP 平台。
11. 如有异常或者错误发生，异常处理框架将记录异常消息，经过处理，填充适当的响应消息。
12. 消息转化模块接下来把服务提供者的消息格式，经过聚合拼接等操作，转化为服务消费者的消息格式。
13. 性能监控模块被再次调用，以记录接受终端服务提供者的应答信息的时间点。
14. 应答信息返回给异构信息系统（服务消费者）。

15. 性能监控模块被再次调用，以记录 HIP 平台发送应达信息给异构信息系统的的时间点。

4.3 HIP 框架服务实现

本章对 HIP 框架服务的各个模块实现进行系统的阐述和解释，针对其中不同的实现方式也进行讨论。

4.3.1 HIP 异常处理组件

HIP 平台的业务组件遵循一个一致的模式来处理异常和错误。这样异常处理技术的主要好处是减少了服务消费者和服务提供者之间的耦合性。

当异构系统（服务消费者）调用 HIP 平台服务时，调用跨越了三层，分别为 1) SOA 管理层，2) HIP 平台服务层，3) 服务提供者层。异常可能在这三层中发生。所以 HIP 异常处理组件分别针对这三个层次，不同粒度的异常提供了特定的定义方式和处理方式，以分别处理特定的异常，并进行日志归类。异常类别如下。

- 系统异常 - 举例：安全验证失败、消息验证失败等。
- 业务异常 - 举例：业务逻辑错误等。
- 超时异常 - 举例：在特定时间框架内，服务未能按时返回。

SOA 管理层

本层验证传入的 SOAP 请求，并且进行授权检验。如果这些安全性的问题导致安全检验失败，本层会产生 SOAP 失败消息，这种异常属于系统异常的一种。

HIP 服务层

如果 HIP 平台的服务发生异常，举个例子，当进行消息验证和转化聚集时，发生了异常或者错误，HIP 平台会根据不同情况，抛出业务异常和系统异常。

服务提供者层

如果异常发生在服务提供者这个层次，根据不同的情况，可能为系统异常或者业务异常。当系统异常发生时，本层返回 SOAP 错误消息。当业务异常发生时，本层会返回具体的消息，指出那些业务行为存在异常。

超时异常

超时异常发生在以下情况中。

服务提供者超时：如果服务提供者未能在希望的时间内返回正确的消息，超时异常则会被捕捉。这种异常与服务提供者的负载均衡有关，可能反馈到配置信息。

HIP 平台服务超时：HIP 的 SOA 层服务如发生延时，这种异常就会被抛出。

HIP 的异常处理机制

HIP 的异常处理组件根据不同的异常发生层次和异常类型进行处理。异常处理组件可以嵌入任意 HIP 平台运行的消息流的处理当中。

当系统异常在 HIP 服务层或者服务提供者层发生时，特定的 SOAP 错误消息会被创建，并返回至服务的调用者。

当业务异常发生时，HIP 服务定义了特定的消息结构，基于获得的错误信息，构建错误报文。消息格式的标签为“ErrorOrWarningInfo”，包含了以下元素：

- AcknowledgeIndicator 为异常指示元素 - “S”代表消息无异常，“E”代表含有错误信息，“W”代表含有警告信息。
- ErrorCode - 包含了具体的错误代码。
- SourceOfError - 阐述了错误来源的信息，错误是由于服务提供者返回的错误，还是由于 HIP 平台服务所产生的错误。
- FreeText - 具体地阐述相关的错误和异常。

当超时异常发生时，服务超时的返回消息会以企业级 SOAP 错误的形式，发送给服务消费者。当然最终的错误报文会由 HIP 平台拼接生成，并且错误信息会被日志模块捕获。

如果是由于负载均衡等配置原因所导致的超时，可以在核心服务中进行配置调优。

4.3.2 会话管理模块

会话管理模块负责在 HIP 环境中，不同类型会话的管理。当异构系统调用 HIP 平台服务，发送请求消息时，请求消息经过授权和验证，会生成会话的唯一标识——会话 ID。基于服务消费者的不同需求和业务逻辑，HIP 平台服务提供了不同种类的会话：

- 泛用会话

- 预定义证书的会话
- 客户端证书的会话
- 上下文会话

下面的流程图描述了不同会话场景的一个简单的视图，并简单阐述了流程。

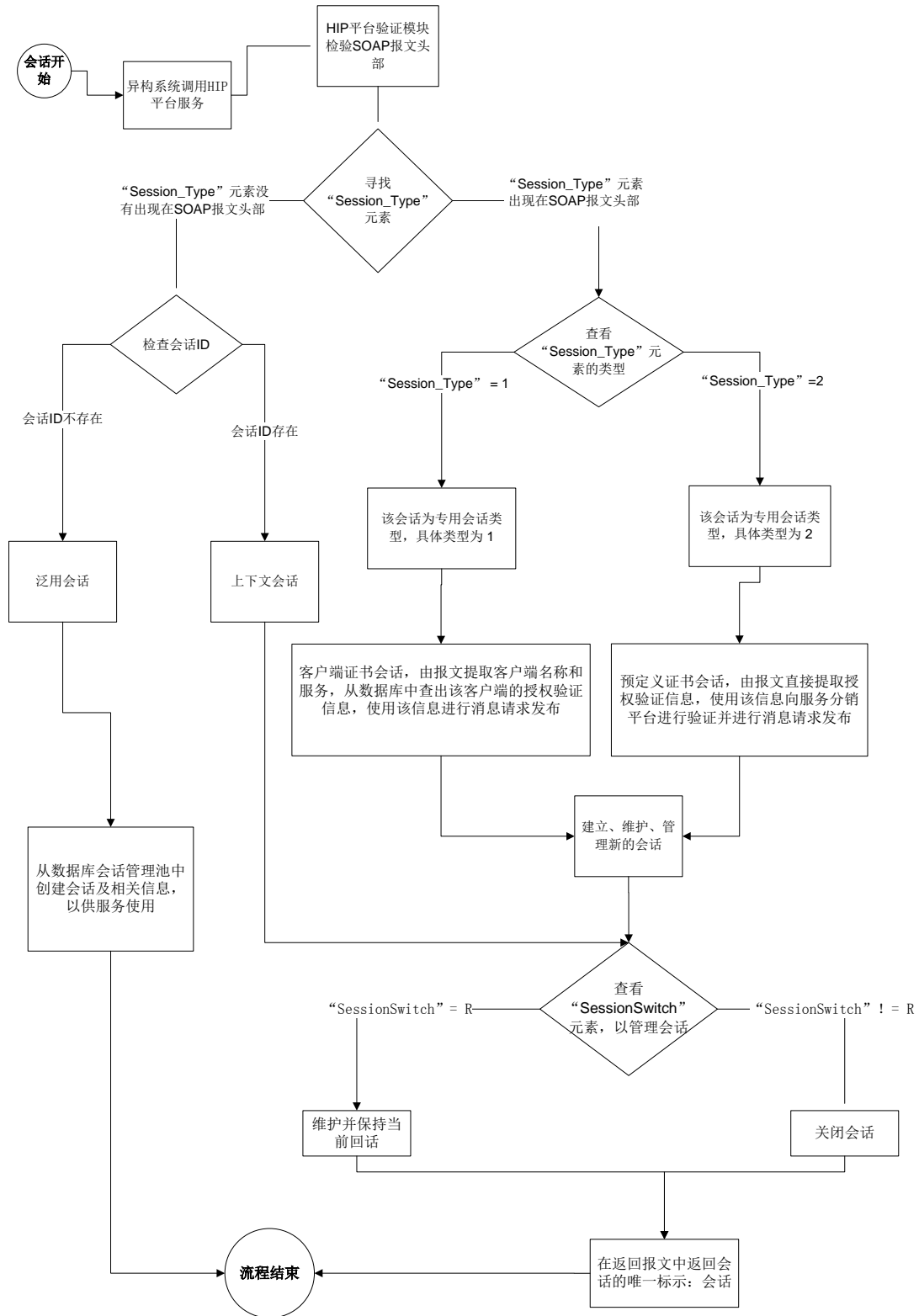


图 4.3 HIP 会话管理流程

1. 泛用会话

这种会话的主要应用场景，是当异构系统向 HIP 服务平台发出只读请求时。比如“查询航班时间服务”、“查询航班编排服务”、“查询特定航班的旅客信息”等服务。当 HIP 平台接受此类型的消息请求时，首先从已有的泛用会话池中选取合适的泛用会话，然后把会话 ID 赋予消息请求，再进行相关处理。

2. 定制会话

对于一些特定的操作，比如“修改”某些特定信息，异构数据信息系统需要安装一定顺序的两次或者多次调用 HIP 平台的信息服务。比如先进行“查询旅客乘机信息”，再进行“旅客乘机信息更新修改”。

在这样的场景，根据实际需求，HIP 平台允许客户系统定制该会话。当定制会话在首次消息请求中建立时，会话唯一标示会被赋予给客户系统，并通过应答消息返回。在以后的一次或者多次消息请求中，客户服务系统只需要复用相同的会话 ID 进行消息请求。

定制会话有两种：

1. 一种定制会话包含预定义的服务分销平台的验证信息。

在这种情况下，HIP 平台在内部数据库维护了一个分销平台验证信息的安全列表。通过不同的异构系统名称、应用名称，映射到对应的安全验证信息。当异构系统发送包含服务分销平台验证信息时，系统名称和应用程序名称同时也会在报文中存储。HIP 平台服务提取这些信息，并根据系统名称和应用程序名称进行映射和校验。当检验通过时，HIP 平台建立与服务分销平台的会话，并进行接下来的消息传输服务。这种形式的定制会话，被称作类型 1 的定制会话。

2. 另外一种定制会话包含客户端的验证信息。

在这种情况下，客户端系统所发送的消息请求含有该客户端的验证消息，比如该客户端的名称、用户名、用户组以及密钥等信息。HIP 平台将直接调用服务分销平台去验证用户信息。当验证通过后，HIP 平台建立与服务分销平台的会话，并进行接下来的消息传输服务。这种形式的定制会话，被称作类型 2 的定制会话。

3. 上下文会话

根据上文的阐述，定制会话，不论是类型 1 还是类型 2，客户系统都会发送额外的信息元素以完成会话，除去特定的验证信息元素，还存在其他元素用以管理会话的状态，这些特定的元素我们称作标示元素。“SessionSwitch”就是

这样一组标示元素。当“SessionSwitch”标示为“R”时，代表 HIP 平台需要管理维持该会话，并将会话 ID 返回给调用者以便于继续基于现在上下文的会话。客户信息系统将继续使用相同的会话 ID 进行信息传输操作。如果“SessionSwitch”的标示不为“R”，那代表会话可以终止，会话资源可以回收。

报文标示的例子如下：

```
<hip:SessionType
xmlns:hip="http://www.gbcs.com/Extensions/">1</hip:SessionType>
  <hip:Session soapenv:mustUnderstand="0"
xmlns:hip="http://www.gbcs.com/Extensions/">
    <hip:SessionSwitch>OP</hip:SessionSwitch>
  </hip:Session>
```

4.3.3 审计服务模块

审计服务模块审计请求和应答消息。这些审计消息存储在 HIP 的数据仓库中。

审计服务设计为消息流中的可复用模块。在消息中间件中，可作为消息流的一个子流，被引入消息流进行审计分析处理。

当服务需要审计处理时，审计子流被加入到业务服务流之中。子流提取存储审计信息，并通过 Message Queue 发送到消息队列。该消息队列会异步地被读取调用，把审计信息存储到数据库当中。

所以审计服务其实为两段服务，实时提取信息和异步存储信息。

审计服务主要可用两种方式实现，优势和劣势如下。

表 4.1 审计服务实现方式

实现方法	优点	缺点
ESQL 方法	1. 可以灵活实现，易于编程。	1. 审计消息格式不明确。 2. 同步的收集存储会降低系统性能。
审计子流	1. 审计消息格式明确。 2. 异步存储。 3. 可复用该模块，增加开发效率。	1. 审计数据必须在调用子流之前构造准备完毕。

步骤一：

先下面的消息流中，提取请求消息，并建立审计消息。

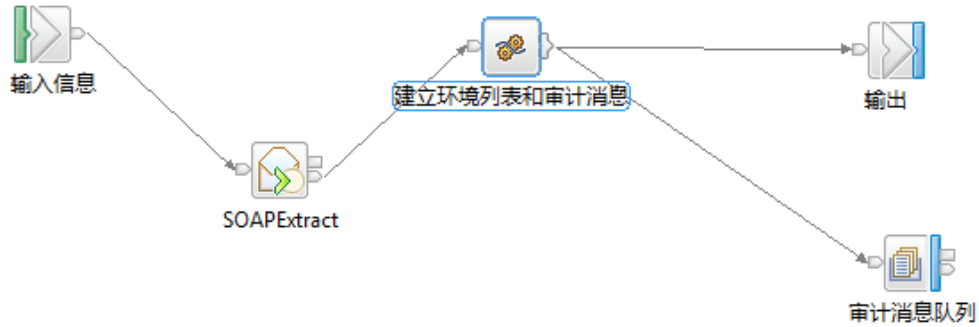


图 4.4 审计服务消息流 1

步骤二：

在下面消息流中，从消息队列（MQ）中提取审计消息并储存到数据库。

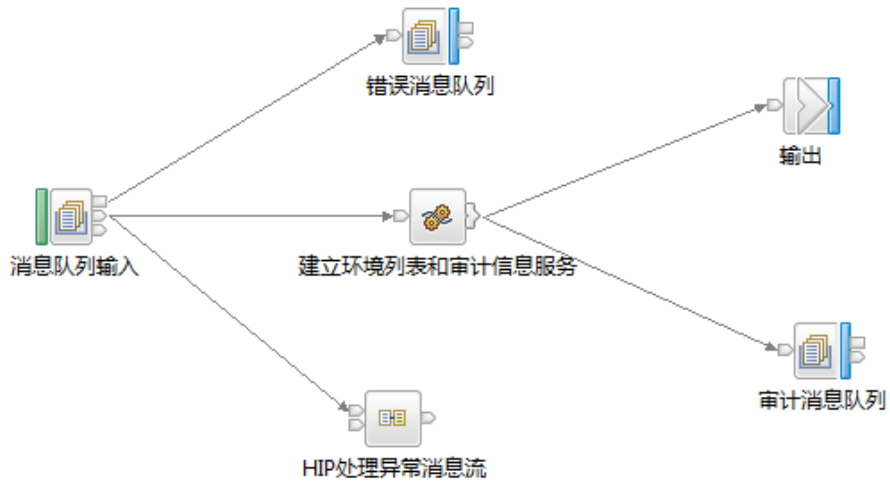


图 4.5 审计服务消息流 2

4.3.4 性能管理服务模块

HIP 框架服务支持对于性能管理数据的收集分析。HIP 平台服务性能管理分析，主要是根据各个节点调用服务的时间点来进行统计分析。

根据具体的业务逻辑，性能管理可以灵活地被引入或者关闭。

基本的性能管理分析基于四个服务时间点的数据。

- Record_Time_HIP_IN：当 HIP 平台服务被引入的时间点。
- Record_Time_HIP_OUT：当 HIP 平台服务返回应答消息给服务调用者的时间点。
- Record_Time_End_IN：服务请求发送给服务分销平台的时间点。
- Record_Time_End_OUT - 服务分销平台应答返回的时间点。

所以，基本的性能数据计算公式如下所示：

- 统计的服务运行时间 = Record_Time_HIP_OUT - Record_Time_HIP_IN。
- 统计的服务分销平台处理时间 = Record_Time_End_OUT - Record_Time_End_IN。

所以，在 HIP 平台的总的消息解析和转化时间 = 统计的服务运行时间 - 统计的服务分销平台处理时间。

消息流操作：

- HIP 平台服务把消息放入性能管理消息队列。
- 从异构系统发出消息请求的时间戳，返回异构系统的应答消息时间戳，以及服务分销平台的请求和应答时间戳。性能管理模块抽取关节时间戳进行记录，并根据时间戳计算性能信息。

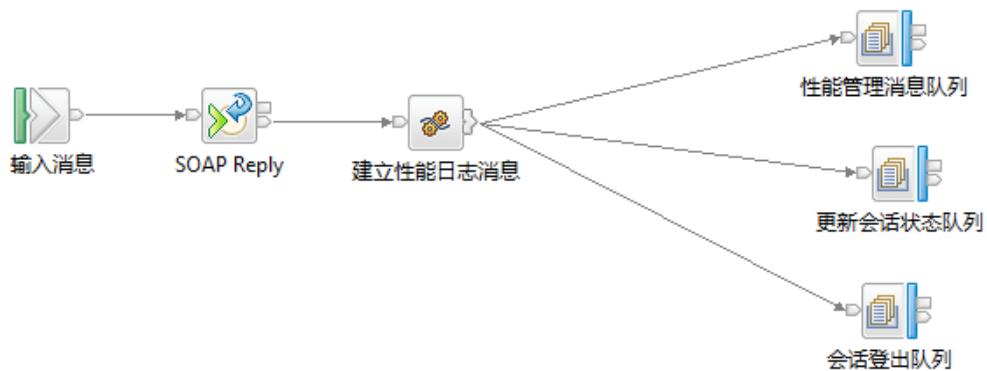


图 4.6 性能管理服务消息流

4.3.5 日志管理服务

日志管理服务模块对所有模块和消息流提供标准的日志记录功能。

该日志记录的格式基于 XML，既易于运维工程师阅读，也可被系统管理软件解析进行分析处理。

定义 HIP 平台服务的日志服务，应该包括记录哪些潜在的消息到日志服务，传递什么样的信息。

每个服务请求会由消息请求者定义设置日志记录级别。每个服务请求也会有一个默认的日志级别，该日志级别将被应用到消息流，当存在预设值和消息请求者设定的日志级别时，更高的级别将覆盖较低的级别。日志管理级别在发起信息服务时，在消息头进行初始化和赋值，在整个消息流过程中不会更改。

日志的管理级别等设置信息主要有以下三种实现方式。

表 4.2 日志管理服务实现方式

实现方法	方法内容	综合考量
中间件外部变量	使用中间件的外部变量，然后在消息流、部署报中定义使用外部变量。	这个方法需要在中间件定义外部变量，而且日志功能的不同级别和启用关闭日志都需要定义外部变量。 这个方法有着以下副作用： <ul style="list-style-type: none"> • 如果外部变量变化，消息流需要重启使得变量生效。 • 对多级的日志支持不好。
系统变量	使用系统环境变量	这个方法基于系统环境变量，而且日志功能的不同级别和启用关闭日志都需要定义外部变量。 这个方法有着以下副作用： <ul style="list-style-type: none"> • 如果外部变量变化，中间件服务需要重启使得变量生效。 • 对多级的日志支持不好。
属性文件	在特定的属性文件定义不同的日志级别	这个方法在属性文件中定义日志级别。用户可以直接更改属性文件以设置日志的级别。 <ul style="list-style-type: none"> • 这个方法提供了对于日志文件的灵活实现。可以灵活动态地更改日志级别以及开启、关闭日志功能。

最后根据需求，我们选择了属性文件方式实现日志管理服务。

在属性文件当中，属性的设置结构如下。

- HIPInLogLevel - INFO : HIP 平台的服务传入消息的日志级别设置为 INFO。
- HIPOutLogLevel- INFO : HIP 平台的服务发出消息的日志级别设置为 INFO。
- ProInLogLevel- DEBUG : 终端服务提供者传入消息的日志级别设置为 DEBUG。
- ProOutLogLevel - DEBUG : 终端服务提供者传出消息的日志级别设置为 DEBUG。
- LogLg - 2000 : 日志文件的长度为 2000。

这里的日志级别分类和 log4j 类似分为：

```
static Level DEBUG
```

DEBUG Level 指出细粒度信息事件对调试应用程序是非常有帮助的。

```
static Level INFO
```

INFO level 表明消息在粗粒度级别上突出强调应用程序的运行过程。

```
static Level WARN
```

WARN level 表明会出现潜在错误的情形。

static Level ERROR

ERROR level 指出虽然发生错误事件，但仍然不影响系统的继续运行。

static Level FATAL

FATAL level 指出每个严重的错误事件将会导致应用程序的退出。

4.3.6 异常管理子模块

这里的异常管理模块负责处理所有 HIP 平台传输解析中的异常。

当消息中间件解析请求消息和应答消息产生异常时，异常信息被捕获到异常处理模块集中处理，并提供日志机制进行错误追踪。

异常管理模块将进行以下处理：

- 从消息中间件的异常错误队列中读取异常消息，并推送存储到 HIP 平台的 ESB 数据仓库。
- 手动触发以重现异常错误消息。
- 计划触发以重现异常错误消息。
- 如果业务逻辑存在对于某种异常处理的消息流，异常将被发送到特定消息流进行处理。

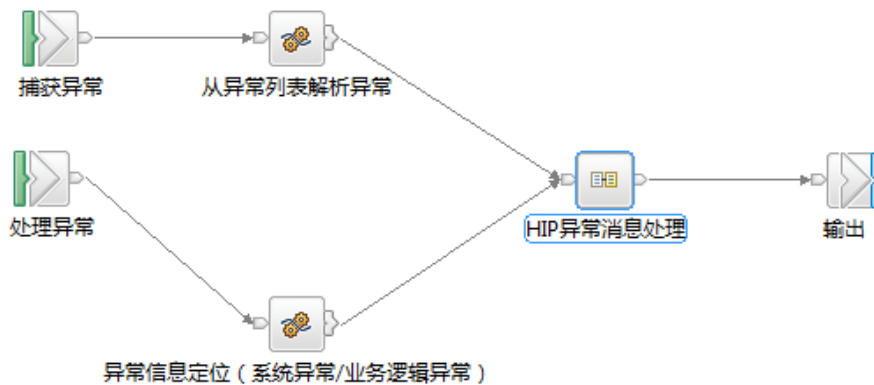


图 4.7 异常管理服务消息流 1



图 4.8 异常管理服务消息流 2

4.3.7 参考数据服务

参考数据服务管理者其他服务所需要的参考数据。参考数据的内容如下：

- HIP 平台服务的配置信息。
- 统一泛用的参考消息，把 HIP 平台内部的一些编码转化成可读的文件文档描述。
- 参考数据服务的首要目的是支持 HIP 平台服务对参考数据进行统一的管理和配置。

这个模块为平台控制设计，所以平台运行的临时数据也是管理的主要内容。临时数据包括消息中间件和外部系统的通信数据，字典表的映射数据，审计顺序和事件等。

4.4 HIP 数据服务实现

4.4.1 电子数据交换解析服务

电子数据交换解析服务包含以下信息。

解析模块

EDIFACT 规范制定的方式和它的应用范围，确定了它在数据表达方面的特点。首先，EDIFACT 的规范是非常全面，覆盖范围包括各个可能使用该规范的行业系统，基本上是只要有某种特定需求，就会在规范中编写制定相应的功能段。这就决定了在现实中每个用户都只会使用 EDIFACT 规范的一个缩减版。所以最终的报文是否需要按 EDIFACT 的规范来校验，并不是必须的。

所以 EDI 应用的设计方案一般都是给每个客户做一个定制解析/映射。

那么映射该如何做？EDIFACT 的结构规范是 SEGMENT GROUP => SEGMENT => DATA ELEMENT，这里略掉了 COMPOSITE DATA ELEMENT，因为我觉得它和 DATA ELEMENT 的处理方式很类似。

最初接触到 EDI 的时候，解析模块是针对 SEGMENT 来做的，但后来发现这样的设计思路是有问题的。首先自然而然地考虑到每个 SEGMENT 都会在报文的不同地方做不同的用途，所以每个 SEGMENT 的处理就必须是通用的，这样就在 SEGMENT 这个层面上又回到了 EDIFACT 的标准规范中去了，而实际应用中就会发现，SEGMENT 的标准规范也是没有什么用的。而且，在业务需求的应用中，想根据一段 EDI 报文创建一个对象或者数据库中的一个记录行，最终还是会在 SEGMENT GROUP 的层面上体现。

所以，报文的解析/映射应当放在 SEGMENT GROUP 的层面上进行。使用 FACTORY 模式/BUILDER 模式/STRATGY 模式来解决不同 SEGMENT GROUP 的处理。

那么，如果使用一个 BUILDER 类来根据一段报文创建一个对象，这个 BUILDER 类的粒度该有多大？EDIFACT 中 SEGMENT GROUP 的层可能很深，GROUP 套 GROUP，有个 10 层绝对是很常见的，如果按每个 GROUP 来设计一个 BUILDER 类，那么这些 BUILDER 类的层次就很难看了。如果按业务对象或者数据库表结构来设计 BUILDER 类，会造成 BUILDER 类过于庞大。有些报文映射到数据库可能就是 1 到 2 张表，而报文却有 8~9 层的结构，BUILDER 类就必须分解。

所以本文建议自顶而下的分解，先按类结构或库表结构设计 BUILDER 类，然后逐步分解，直到 BUILDER 类的大小合适。

在 HIP 平台服务的业务对象的设计中，EDIFACT 标准只作为参考，设计过程按业务需求来设计，从需求中分析出业务数据模型，再建立对象

以及相关的库表结构,然后从 EDIFACT 报文中找到相关的字段来建立对象。这就是 MAPPING 的思路。

关键在于, EDIFACT 规范中的循环结构和多层次的嵌套在业务类的模型里是不应该出现的。

验证模块

验证模块服务验证信息符合 EDIFACT 标准。该模块检验所有必要的群组、消息段、消息元素,以及检验这些元素的重复次数未超过标准中定义的次数。除此之外,验证模块还会做数据质量的初步测试(数据长度、类型等)以确定与定义相符。所以验证模块可以防止数据库层面出现数据类型不一致等错误,简单的早期检验和错误数据的拒绝可以有效地减少应答时间。验证模块在通用消息树建立后进行验证操作,这样可以有效定位异常,并加速简单消息解析。

适配模块

适配器组件使用观察者模式构建的消息树解析器生成实体,提供所需的输出格式(使用批处理 SQL 的 CSV 或 DB)。实体的结构可以在属性文件中定义的。以下两种类型的适配器,为这个解决方案而设计,并且都可以无缝嵌入框架工作。

数据库适配模块

数据适配模块生成数据库查询信息,通过遍历段(属性文件中定义)节点下的对照组,然后加入批处理 SQL 生成的条目。批处理 SQL 包含多个 SQL INSERT 语句,并通过 JDBC 方式执行 SQL。

文件适配模块

文件适配器产生一个输出文件,根据参照组的节点(在属性文件中定义),转化解析消息段的。然后将所生成的条目添加到一个缓冲区。缓冲区将被一次性的刷新到输出文件。

4.4.2 工作数据仓库

工作数据仓库提供了关系数据库的逻辑模型和物理模型,用以存储服务分销平台推送的业务数据。

数据仓库根据业务逻辑和主题建立。模型包含以下主题：

- 旅客订票信息。
- 库存和清单信息。
- 服务分销平台提供的客户信息。
- 服务分销平台提供的座位信息。
- 服务分销平台提供的行李信息。
- 等等.....

模型的设计使得复杂信息中,各个信息片段、主题、表项之间的关系便于理解分析。根据不同的业务需求,设计定制数据仓库以实现数据的持久化。

4.5 HIP 功能服务

本节首先阐述 HIP 平台功能服务的总体框架和高层设计。

所有功能服务都可以被发布为 Web 服务,以供外部系统调用。在这里我们只选取最基本的用例和场景,不牵涉到复杂的服务编排内容,比如,建立旅客订票信息,查询特定航班特定仓位,查询乘客行李相关信息等。

本节包含以下内容：

- 功能服务的组件模型
- 功能服务目录
- 功能服务对于会话类型的映射
- 功能服务的实现
- 功能服务高层描述和实现

4.5.1 功能服务组件模型

下面就是 HIP 平台功能服务的组件模型。这些服务根据主题和业务逻辑,被分为以下几个子类型：

- 在线预订服务
- 旅客订票服务
- 航班服务

- 费用服务
- 票务服务
- 等等.....

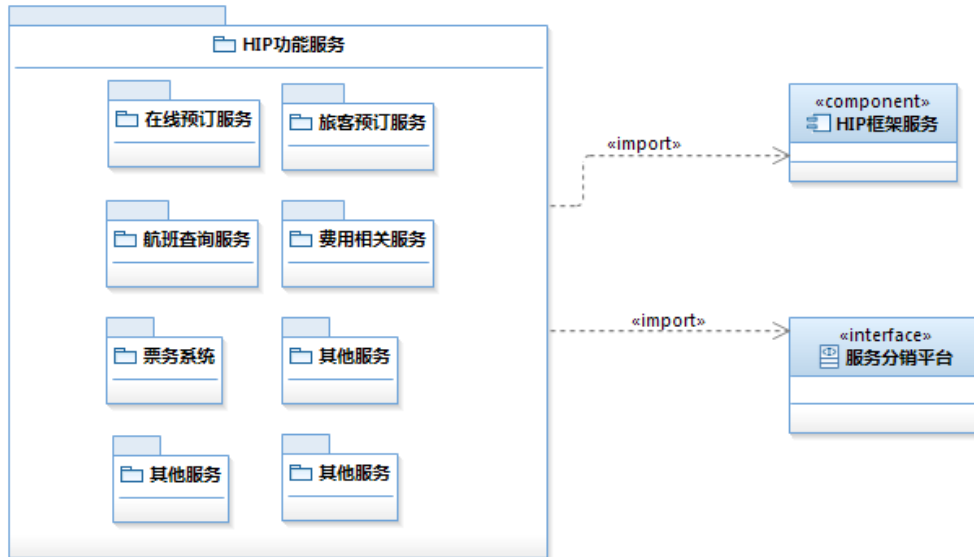


图 4.9 功能服务组件结构

4.5.1.1 组件关系

HIP 平台的每个基本服务都是独立的服务，所以每个基本服务实现时，不会调用其他的功能服务。在基本服务上，会根据业务逻辑和用户需求进行编排构造复杂服务。基于以下 SOA 框架，实现组件和服务的多级框架。

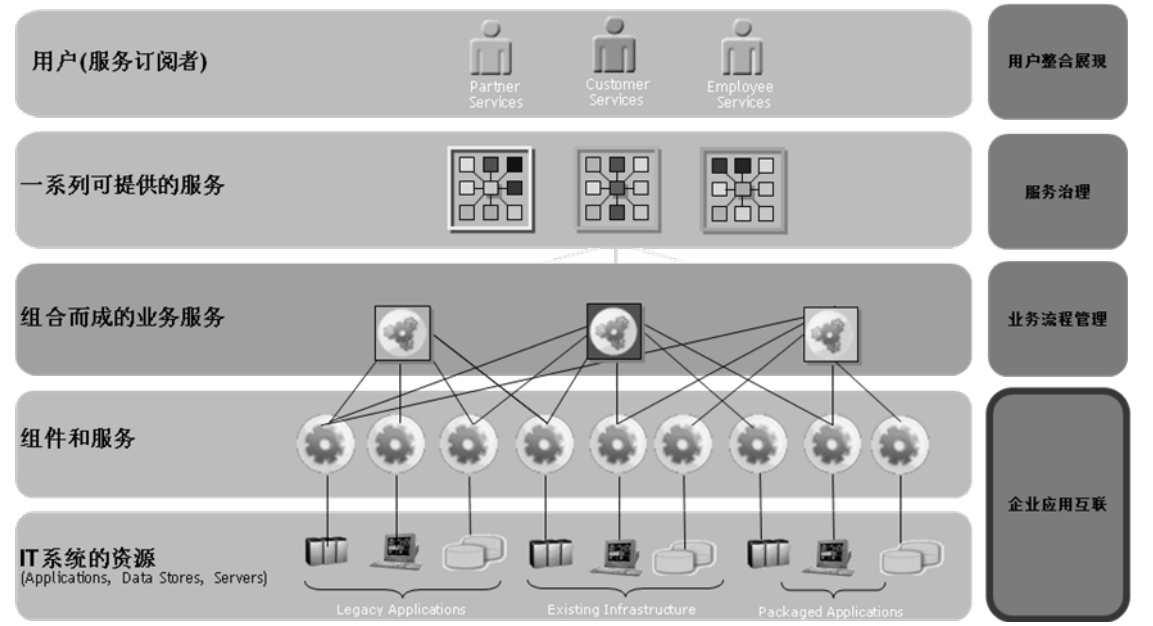


图 4.10 功能服务组件层级架构

组件交互

HIP 平台的服务不但可以调用其他功能性服务从而进行复杂服务的编排，也可以调用框架服务以及直接从终端服务提供者请求信息。

4.5.1.2 服务类别

下面是 HIP 平台所发布的基本功能性服务（业务服务）列表，这里只列举一些常用的基本服务：

表 4.3 基本功能服务列表

HIP 平台常用基本功能性服务列表	
在线预订服务	删除旅客预订信息
	建立旅客预订信息
	查询旅客预订信息
	修改旅客预订信息

旅客订票服务	查询旅程信息
	更改航段信息
	更新预订渠道
	批量更改旅客姓名

航班服务	查询航班信息
	查询航班时刻表
	查询班号共用的航班信息
	查询航班座位信息

费用服务	根据价格查询旅客订票记录
	根据价格选取旅程信息
	获得旅程相关价格

票务服务	清理票务记录
	获取票务销售汇总
	获取票务详细信息
	发放电子机票
	基于价格查询 TST (机票记录)
	修改 TST (机票记录)
	显示 TST (机票记录)

4.5.1.3 组件实现消息流

本节详细描述了 HIP 平台服务系统的实现细节。

HIP 平台服务系统使用 Websphere Message Broker 环境，以消息流的形式实现。服务流程的机制描述如下：

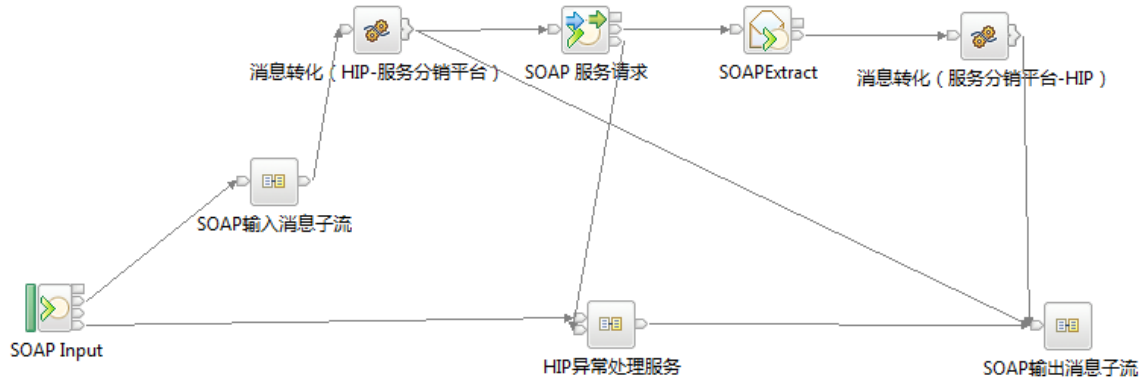


图 4.11 组件实现消息流

异构系统通过传输消息请求给 SOAP INPUT 节点，来调用 HIP 平台提供的 Web 服务。

- SOAP Input Node: 当异构系统使用基于HTTP的SOAP报文引入HIP服务时，该消息请求首先会被SOAP Input节点捕获。SOAP Input节点会根据消息定义的格式和HIP平台的配置，对消息内容进行验证。
- SOAP输入消息子流: 本节点接受消息，并且初始化审计消息和日志消息。并且把审计消息放入审计消息服务队列。审计消息服务队列会被审计消息服务再次调用，异步地插入到数据库中。另外一方面本节点会支持会话验证的消息流，在子流中管理会话。

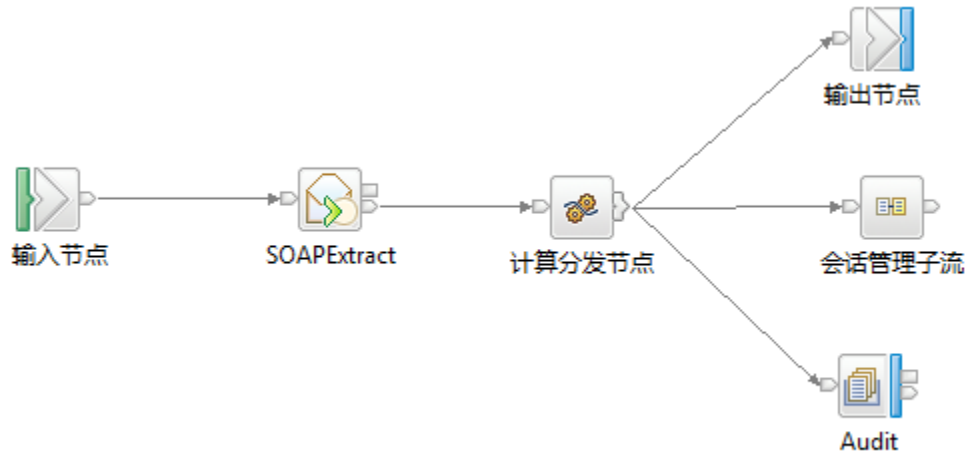


图 4.12 SOAP 输入消息子流

- 消息转化 (HIP-服务分销平台):在本节点, 输入请求消息依照义务需求, 被转化为服务提供者可以接受的格式。
- SOAP 请求:本节点负责向服务提供者发送基于SOAP_HTTP的请求报文, 并接受服务提供者返回的服务应答。
- SOAP Extract:本节点解析SOAP报文, 去除SOAP信封。
- 消息转化 (服务分销平台-HIP):本节点负责把HIP平台服务的消息格式转化为调用平台服务的异构系统所能理解的消息格式。
- SOAP输出消息子流:本节点具体有以下功能。
 - 在返回消息推送给服务分销平台或者写入外部文件前, 记录应答消息, 并进行日志服务。
 - 该节点为消息子流, 子流含有SOAP reply节点, 应答消息会被该节点返回给调用者, 然后应答消息会在消息格式的转化进行后, 被存入数据库。
 - 抓取服务应答时间的时间戳。
 - 将性能数据放入队列, 该消息会在将来被HIP性能服务获取, 存入数据库或平面文件进行分析处理。
 - 调用会话管理服务, 对于上下文会话, 把会话注销信息放入会话注销消息队列。如果 SessionSwitch 设置不为“R”, 那么会话的安全注

销服务会异步的对会话注销队列中的会话进行终止注销，并把记录存入数据库。

- 对于泛用会话，则会放入会话信息更新队列，泛用会话更新服务稍后会从队列中获取泛用会话信息，然后进行会话池中的维护和更新，以确定活跃泛用会话的数量在一个稳定可用的值。

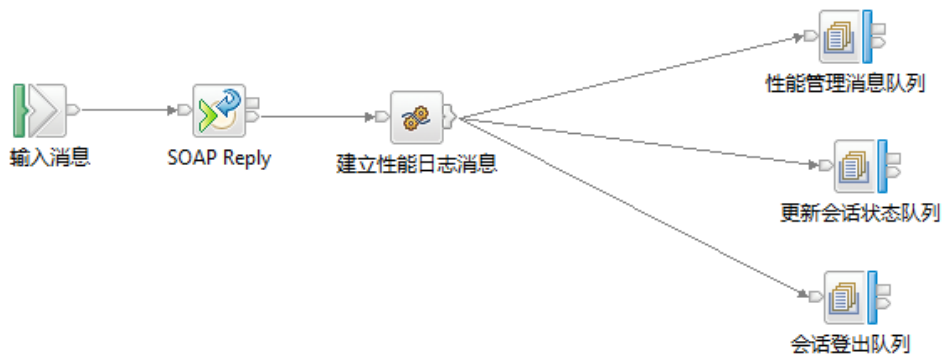


图 4.13 SOAP 输出消息子流

- HIP异常处理服务 :如果消息流中有任何错误发生，异常处理子流会向消息请求者返回具体错误信息。

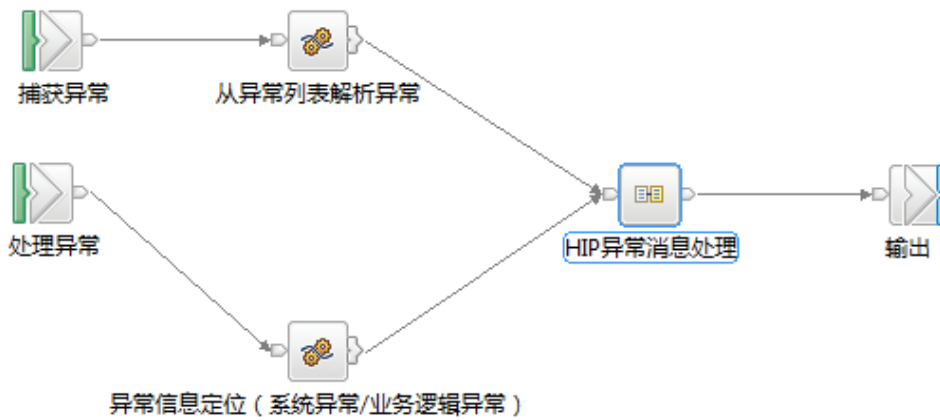


图 4.14 异常处理流程

4.5.2 在线预定系统

在线预订系统包括的原子服务举例如下。

建立旅客订票信息

建立旅客订票信息服务是基于 SOA 的基本服务，可以被异构系统调用，以建立新的旅客订票信息。

为了建立新的旅客信息，客户端应用需要提供以下信息：

- 旅客姓名数据 - 旅客姓名、姓氏、头衔。
- 订票信息 - 航班号、离港日期、始发地、仓位以及操作码。
- 可选信息 - 特殊服务需求，比如特殊餐饮服务、护照和 VISA 服务；以及其他相关服务信息，比如紧急联系信息。

删除旅客订票信息

删除旅客订票信息服务是基于 SOA 的基本服务，可以被异构系统调用，以删除已经存在的旅客订票信息。

为了建立新的旅客信息，客户端应用需要提供以下信息：

- RLOC (record locator, 旅客记录信息) 旅客订票信息的键值。
- 所有需要删除的乘客姓名 - 姓氏、名字、头衔 (可选, 非必须)。
- 订票信息段信息 - 航班号、离港日期、出发的-目的的、仓位级别、位子数量、操作码 (可选, 非必须)。

查询旅客订票信息

查询旅客订票信息服务是基于 SOA 的基本服务，可以被异构系统调用，以查询旅客订票信息的详细情况。

这个服务的主要目的是提供旅客订票信息的详细情况。服务消费者发出请求消息，并获得详细的旅客订票信息。不同异构系统发出的请求信息可能有所不同。但是根据旅客订票信息的 RLOC 可以唯一确定订票详细信息。通过旅客订票信息的键值，获得响应的消息应答。

为了查询旅客详细的订票信息，客户端应用需要提供以下信息：

- 旅客订票信息键值 RLOC

修改旅客订票信息

修改旅客订票信息服务是基于 SOA 的基本服务，可以被异构系统调用，以修改已存在的旅客订票信息的详细情况。

这个基本服务的目的是增加、更新和删除旅客订票信息中的元素。可以通过以下两种方式实现：

根据服务请求，服务消费者需要提供整个存在的旅客订票信息。在获取旅客订票信息后，服务消费者可以发送消息请求以修改旅客订票信息。另外一方面，旅客可以直接通过发送带有旅客订票信息键值 (RLOC) 的消息，请求直接修改旅客订票信息。

这个服务提供了 5 功能：

- 在已有的旅客订票记录中加入元素
- 更新已有的旅客订票记录中的元素
- 删除已有的旅客订票记录中的元素
- 提交旅客订票信息更改

- 回滚旅客订票信息更改

注意：为了在已有的旅客订票信息中调用更新或者删除元素，服务消费者需要根据业务需求，在消息请求中，提供标示性的消息元素。具体的消息元素由异构系统决定。而更新元素的本质是一系列的删除和加入信息元素。

4.5.3 旅客预订系统

旅客预订系统包括以下原子服务举例如下。

查询旅程信息

查询旅客旅程信息服务是基于 SOA 的基本服务，可以被异构系统调用，以查询旅程相关信息。

这个服务的目的在于根据输入的常旅客信息或者用户信息，提供该旅客的所有旅程信息。

为了查询旅客旅程信息，客户端应用需要提供以下信息：

- 常旅号/客户账号
- 公司代码（可选，非必须）

如果以上服务没有提供相应的必要元素，则会根据必要元素的缺失，返回相应的错误信息。

更改航段信息

更改航段信息是基于SOA的基本服务，可以被不同异构系统调用，以更改航段信息。

本服务的目的在于更改与旅客订票信息相关的一个或多个航段信息。

为了更改航段信息，客户端应用需要提供以下信息：

- 应用程序名 - 异构系统中发出服务请求的应用系统。
- 服务分销平台标识 - 服务分销平台对外部提供信息服务时，所需的标识。
- 取回标识 - 是否把更改的航段信息取回 HIP 平台，以推送到相关订阅异构系统。

更改预订票务渠道

更改预订票务渠道是基于SOA的基本服务，可以被不同异构系统调用，以更改预订票务渠道。

为了更改航段信息，客户端应用需要提供以下信息：

- 应用程序名 - 异构系统中发出服务请求的应用系统。
- 服务分销平台标识 - 服务分销平台对外部提供信息服务时，所需的标识。
- 取回标识 - 是否把更改的航段信息取回 HIP 平台，以推送到相关订阅异构系统。
- 渠道标识 - 标识相关票务渠道。

批量更新旅客姓名

批量更新旅客姓名是基于SOA的基本服务，可以被不同异构系统调用，以批量更新旅客的姓名。

这个服务基于特定的需求，在特定的群组更新旅客姓名，对其他预订信息没有更改。主要支持批量预订系统的服务请求。

4.5.4 航班查询系统

航班查询系统包含以下几个基本服务：

查询航班信息

查询航班信息是基于SOA的基本服务，可以被不同异构系统调用，以批量查询航班的信息。

服务消费者可以使用该服务取得特定航班的可用仓位信息。所以，仓位信息字段可以在请求信息中体现。仓位的大类包括：经济舱，商务舱，头等舱等，以下细分很多子类。例如：优质商务舱，特价折扣经济舱等。仓位信息在专门的字段进行描述，并且以二级目录的形式进行分类描述。

除此之外，必须在该服务中加入时间粒度，选取一定的时间段，以返回相应的可用仓位信息。

服务会批量获取航班信息，进行每条信息的验证，返回每条信息的警告、异常等信息。如果其中出现一处错误，整个消息组将会报错，并封装异常信息，返回给调用者。

查询航班时刻表

查询航班时刻表是基于SOA的基本服务，可以被不同异构系统调用，以查询航班时刻信息和航段信息。

这个服务主要有两个目的：

- 查询航班安排
- 查询航段信息

查询航班安排：

为了查询旅客旅程信息，客户端应用需要提供以下信息：

- 离港日期
- 离港机场
- 到达机场
- 承运人代码（可选、非必须）

需要注意的是，根据三大航空联盟（天合联盟 SKYTEAM，星空联盟 STAR ALLIANCE，寰宇一家 ONE WORLD）的不同要求，这个服务需要定制特定消息结构。

最终航班安排的结果由输入参数的出发地点、目的地以及时间维度共同确定。

查询航段信息

查询航段信息需要提供航班的基本信息。另外，时间维度的数据也是必需的。在消息报文中加入以上两者，就会返回限定条件的航段信息和信息组。

- 离港日期
- 离港机场
- 到达机场
- 承运人代码（可选、非必须）
- 航班号

服务将取得多组信息，每组信息都会返回成功确认、警告或者失败信息。如果其中有一组发生错误，或者其中出现一处错误，整个消息组将会报错，并封装异常信息，返回给调用者。

查询班号共用可定航班信息

查询可定航班信息是基于SOA的基本服务，可以被不同异构系统调用，以批量更新旅客的姓名。

为了查询旅客旅程信息，客户端应用需要提供以下信息：

- 所有班号共用航段
- 所有共用的班号，以及内部的航段

***班号共用**

(Code-sharing) 又称共挂班号, 在我国又称之为“代码共享”, 主要应用于多家航空公司共同对某一航段进行经营的过程中。

例如泛美航空与德国汉莎航空签定了“纽约-柏林”班号共享的合约, 约定某时段的 F123 班次与 H321 班次为班号共享班次, 实际上航班及机组成员都由泛美航空提供。两家航空公司仍各自以各自的班次号码卖机票, 泛美航空在这班次的座位, 会有一部分划定给汉莎航空来发售。旅客买了汉莎航空的机票后, 到了机场, 若汉莎航空未在该机场设有报到柜台, 则要到泛美航空的柜台办理报到划位手续; 若汉莎航空设有报到柜台, 则直接到汉莎航空的柜台报到划位。简而言之, 这相当于汉莎航空购买了泛美航空的一系列服务。

查询座次表

查询座次表服务是基于 SOA 的基本服务, 可以被不同异构系统调用, 根据相应的航段和仓位, 返回相应的座次表的结果。

服务调用者至少需要提供航班特定的航段, 以获得座次表信息。服务请求者的请求信息至少包括航班号、离港日期、出发地、目的地、仓位信息、服务的航空系统。然后获取具体的座次表信息, 包括座位统计信息、乘坐率、仓位容量等信息。

4.5.5 费用系统

费用系统服务提供的原子服务举例如下。

根据价格查询旅客订票记录

根据价格查询旅客订票记录是基于SOA的基本服务, 可以被不同异构系统调用, 返回根据价格查询旅客订票记录的结果, 最终返回含有税率和其他费用信息的旅客订票记录。

根据价格选取旅程信息

根据价格选取旅程信息是基于SOA的基本服务, 可以被不同异构系统调用, 返回根据价格选取旅程信息记录的结果, 最终根据消息请求中的价格或价格区段, 返回相应的旅程信息。

获得旅程相关价格

获得旅程相关价格是基于 SOA 的基本服务，可以被不同异构系统调用，返回获得旅程相关价格的结果。需要注意的是，这里的旅程价格信息不需要提供旅客订票记录。

4.5.6 票务系统服务

票务系统服务提供原子服务举例如下。

整理票务记录

整理票务记录是基于SOA的基本服务，可以被不同异构系统调用，以整理票务记录。

整理票务记录服务根据特定的出票渠道，整理并推送票务记录。对所有订阅该服务的异构系统，HIP服务平台会进行数据的推送。

查询票务销售汇总

查询票务销售汇总是基于SOA的基本服务，可以被不同异构系统调用，以查询售票的汇总信息。

服务请求者可以调用HIP服务平台的服务已获得每日的票务销售汇总信息。本服务在整理票务记录前被调用，对所有订阅该服务的异构系统，HIP服务平台会进行数据的推送。

查询票务基本信息

查询票务基本信息是基于SOA的基本服务，可以被不同异构系统调用，以根据航线查询电子售票的基本信息。

服务请求者发送请求给HIP异构服务平台，请求包含异构系统中的应用程序名称，经过验证的机票编码和航班号（可选、非必须）。HIP将查询的结构返回给服务调用者。

发放电子票服务

发放电子票服务是基于SOA的基本服务，可以被不同异构系统调用，用来生成电子票。

服务请求者发送请求给HIP异构服务平台，请求电子机票的相关信息。

基于价格查询 TST（机票记录）

基于价格查询机票记录服务是基于SOA的基本服务，可以被不同异构系统调用，以基于价格查询机票记录。

机票记录与乘客订票信息的键值对应，服务请求者需要提供相应的乘客订票信息以调用HIP异构服务平台的服务。对所有订阅该服务的异构系统，HIP服务平台会进行数据的推送。

修改 TST（机票记录）

修改机票记录服务是基于SOA的基本服务，可以被不同异构系统调用，以修改已经存在的机票记录。

机票记录与乘客订票信息的键值对应，服务请求者需要提供相应的乘客订票信息以调用HIP异构服务平台的服务。

显示 TST（机票记录）

显示机票记录服务是基于SOA的基本服务，可以被不同异构系统调用，以显示相关机票记录。

机票记录与乘客订票信息的键值对应，服务请求者需要提供相应的乘客订票信息以调用HIP异构服务平台的服务。

4.6 数据流转模式和 HIP 服务平台应用场景

本节阐述基于业务系统的最常用的数据流转和推送请求模式。

4.6.1 从异构系统发送到 HIP 平台的服务请求

请求发起者	异构系统（服务消费者）
描述	异构系统发出Web服务请求，调用一个HIP平台预订服务。截获此请求的Web服务的安全网关首先进行认证、授权并随后转发此请求到HIP平台的具体服务。
传输通道	HTTP(s)。 HIP的预订服务对外发布为Web服务。 请求调用使用SOAP/HTTP(s)。
数据模式	包含XML的SOAP消息。
服务类型	同步请求和应答。
质量级别	需要基于客户的实际需求制定。

第 4 章 异构信息服务平台核心分析

用户数量	需要基于客户的实际需求制定。
事务数量和频率	需要基于客户的实际需求制定。
数据量	需要基于客户的实际需求制定。

4.6.2 从 HIP 平台发送到服务分销平台（终端服务提供者）的请求

请求发起者	HIP平台
描述	由HIP平台的服务发送请求到服务分销平台。 注意：当请求发起的初始源为异构系统时，HIP平台解析服务请求并进行数据转换，接着向服务终端提供者发出服务请求，这个过程可以在相同的数据流中实现。
传输通道	HTTP(s)。 服务分销平台发布自己提供的服务为Web服务。HIP平台发出的请求使用SOAP/HTTP(s)。
数据模式	包含XML的SOAP消息。
服务类型	同步请求和应答。
质量级别	需要基于客户的实际需求制定。
用户数量	需要基于客户的实际需求制定。
实物数量和频率	需要基于客户的实际需求制定。
数据量	需要基于客户的实际需求制定。

4.6.3 从服务分销平台（终端服务提供者）返回 HIP 平台的服务应答

应答返回者	服务分销平台
描述	服务分销系统向HIP平台服务返回Web服务应答。在HIP平台，将由向服务分销系统提出服务请求的功能服务来接受应答。
传输通道	HTTP(s)。 使用SOAP/HTTP(s)的Web服务应答。
数据模式	包含XML的SOAP消息。
服务模式	同步请求和应答。
质量级别	需要基于客户的实际需求制定。
用户数量	需要基于客户的实际需求制定。
实物数量和频率	需要基于客户的实际需求制定。
数据量	需要基于客户的实际需求制定。

4.6.4 从 HIP 平台返回异构系统的服务应答

应答返回者	HIP平台服务
描述	由HIP平台服务返回给异构系统（服务消费者）的应答消息。有异构系统发出请求消息的服务接受该应答消息。
传输通道	HTTP(s)。 使用SOAP/HTTP(s)的Web服务应答。
数据模式	包含XML的SOAP消息。
服务模式	同步请求和应答。
质量级别	需要基于客户的实际需求制定。
用户数量	需要基于客户的实际需求制定。
实物数量和频率	需要基于客户的实际需求制定。
数据量	需要基于客户的实际需求制定。

4.6.5 由服务分销平台推送到 HIP 平台的旅客定座数据

传输发起者	服务分销平台
描述	乘客订票信息以及相关消息由服务分销平台提供，也就是说服务分销平台是服务的提供者。各种异构系统获取旅客订票信息，所以各异构系统为服务的消费者。服务分销平台使用 Message Queue 传输旅客订票消息，该消息被HIP的消息解析模块接受并解析。 服务分销平台发送的消息基于工业标准的EDIFACT数据格式。
传输通道	Websphere MQ.
数据模式	使用MQ负载的工业标准EDIFACT数据格式。
服务模式	1. 实时模式。 2. 由服务分销平台向HIP服务平台的异步数据单向推送。
质量级别	需要基于客户的实际需求制定。
用户数量	需要基于客户的实际需求制定。
实物数量和频率	需要基于客户的实际需求制定。 注意：HIP平台设计支持高并发消息以及高容量数据。
数据量	需要基于客户的实际需求制定。

4.6.6 由 HIP 平台服务推送到异构系统的旅客订票信息

传输发起者	HIP平台服务
描述	HIP平台解析传入包含旅客订票信息的EDIFACT消息，并且检测特定的变更。基于发布/订阅模型，向订阅者（异构系统）推

第 4 章 异构信息服务平台核心分析

	送这些变更信息。
传输通道	Websphere MQ。
数据模式	使用MQ负载的XML数据格式。
服务模式	由HIP服务平台向异构系统的异步数据单向推送。 使用发布/订阅模型。异构系统订阅特定旅客订票信息变化的通知。如果旅客订票信息发生改变，即向需要的异构信息系统发送通知，推送消息。
质量级别	需要基于客户的实际需求制定。
用户数量	需要基于客户的实际需求制定。
实物数量和频率	需要基于客户的实际需求制定。 注意：HIP平台设计支持高并发消息以及高容量数据。
数据量	需要基于客户的实际需求制定。

4.6.7 从服务分销平台向 HIP 平台推送库存清单数据

传输发起者	服务分销平台
描述	服务分销平台向HIP平台推送相关库存清单数据。数据基于标准的工业EDIFACT数据格式。数据通过FTP的方式进行批量推送。
传输通道	FTP / 安全 FTP
数据模式	工业标准EDIFACT数据格式。
服务模式	1. 离线批处理模式。 2. 由服务分销平台向 HIP 平台服务的单项异步推送。
质量级别	需要基于客户的实际需求制定。
用户数量	需要基于客户的实际需求制定。
实物数量和频率	需要基于客户的实际需求制定。 注意：HIP平台设计支持高并发消息以及高容量数据。
数据量	需要基于客户的实际需求制定。

4.6.8 从 HIP 平台服务向异构系统推送库存清单数据

传输发起者	HIP平台服务
描述	HIP平台服务批量解析大量库存清单数据，并且存入平面文件，通过FTP方式推送至异构系统。
传输通道	FTP / 安全 FTP
数据模式	平面文件类型
服务模式	1. 离线批处理模式。 2. 由 HIP 平台服务向异构系统的单项异步推送。

质量级别	需要基于客户的实际需求制定。
用户数量	需要基于客户的实际需求制定。
实物数量和频率	需要基于客户的实际需求制定。
数据量	需要基于客户的实际需求制定。

4.7 本章小结

详细地阐述了 HIP 核心的各个模块。

本章首先描述了 HIP 平台各个模块的整体架构图，以及各个模块之间的信息流交互。之后针对 HIP 的各部分核心分别进行了详细的分析和设计。对于 HIP 的各个框架服务，从功能上加以详述，并使用 IBM Message Broker 设计实现了具体的消息流。对于 HIP 的功能服务，介绍了根据业务需求的功能分类，并挑选具有代表性的功能子服务进行设计实现，同样的，使用 IBM Message Broker 实现了具体的消息流。最后，根据实际业务需求，对消息流转推送的模式和应用场景的实现，举例进行了阐述。

第 5 章 总结与展望

5.1 主要工作

首先本文基于对航空业现状，以及未来发展的分析，总结了航空业界信息化面临的首要问题：信息技术的不断革新，产品的周期缩短，市场竞争日趋激烈，需求的变化愈来愈快导致的应用系统快速嵌入、遗留系统和 IT 资产整合利用问题。而这一问题，是重量级企业及应用在快速信息化进程中，普遍面临的问题，鉴于航空业系统的规模和高复杂度，这个问题在该行业尤为突出。

接下来，本文基于 SOA 构架，以服务化、组件化的观点解决这个问题。首先是分析解决问题的背景和技术，针对 SOA 和消息中间件进行了深入调研理解。对行业相关的系统和 EDI 数据模式优缺点有了较为深入的理解，最终在平台的设计阶段，从服务的角度，有针对性的进行了整合实现。

完成核心知识技术调研后，本文设计 HIP 服务平台的整体架构，把系统分为多个组，对与 HIP 进行交互的外部系统（服务分销平台）和 HIP 的扩展系统（Web 服务网关，服务中间件，消息队列，系统管理和监控，安全和目标服务，服务注册表和存储库等）进行了研讨、优化和设计。

针对 HIP 平台核心服务，以组件化的观点分离了可复用的 HIP 核心服务和 HIP 框架服务，针对业务流程设计实现了 HIP 功能服务和 HIP 信息处理服务等，并构建了 HIP 相关的数据仓库。

本文的解决方案在于不是单单利用 SOA 服务中间件实现整合平台，而是把数据和业务做了一定程度上的剥离，服务完全依赖于 HIP 功能服务，在 IBM Message Broker 的支持下，可以做到服务的可视化编排。而数据部分，对数据的解析和数据的流转也进行了较为详实的设计实现。

本文的另外一个创新点是基于 IBM 成熟的中间件技术，实现了异构系统服务平台的整体架构设计和解决方案，最终完成将针对航空业的可复用异构信息处理平台。

5.2 展望

在进一步的研究中，我们可以研究如下问题：

1. 在超大数据量传输时，如果时效性不是很强，我们可以采取批处理，以文件方式推送数据；但是如果时效性很强，使用 Web Services 势必会影响效率，而使用软件套件或者 ETL 等方式会使得 SOA 整个框架的组件灵活性有所牺牲；所以在极大数据吞吐量的情况下，以数据中心为代表，定制的服务有其用武之地；定制优化底层协议也是捷径之一，在今后的工作和实践中，我们会做进一步探讨和设计。

2. 服务的编排存在一定的问题，航空系统的大型企业，会有众多开发商参与开发服务，真正进行规划设计时，很多服务需要重排。而且大型企业中 workflow 极为复杂，其中业务流程语言 BPEL (Business Process Execution Language) 可以一定程度上简化流程，但是 BPEL 始终和统一建模语言 UML (Unified Modeling Language) 存在差别，特别是在并行事件上，所以实现 HIP (异构信息服务平台) 目前在处理并行复杂 workflow 问题上，存在瓶颈。这也是今后可以探讨的问题。

3. 另外一个发人深思的问题是语意，语意关系是设计 SOA 的核心要素。但是，为针对特定行业(航空企业)的业务流程定义，建立数据模型并编排实施服务应用是一项繁重的任务，单靠 IT 管理人员可能导致需求不能被有效满足，所以必须靠业务人员和 IT 人员的有效沟通合作来完成。使用规范化的 XML 是一个较好的解决方案。

参考文献

- [1] Newcomer E. Understanding SOA with Web Service[M]. 徐涵, 译. 北京: 电子工业出版社, 2006.
- [2] Ardagna D, Mirandola R, Trubian M, et al. Run-time Resource Management in SOA Virtualized Environments[C]//Proc. of QUASOSS'09. Amsterdam, Holland: [s. n.], 2009.
- [3] 邓子云. SOA 实践者说: 分布式环境下的系统集成[M]. 北京: 电子工业出版社, 2010.
- [4] IBM. SOA and Web Services: New to SOA and Web Services[EB/OL]. (2010-11-09). <http://www-128.ibm.com/developer-works/soa>.
- [5] Maha Driss^{1,2}, Yassine Jamoussi¹, Jean-Marc Jézéquel², Henda Hajjami Ben Ghézala¹, A Multi-Perspective Approach for Web Service Composition, "iiWAS 2011 (2011)".
- [6] 李福盛, 基于企业服务总线的 SOA 框架研究与应用, 硕士论文.
- [7] 吕希艳, 张润彤, 基于 SOA 的企业信息资源整合, 中国科技论坛, 2006. 3.
- [8] 喻坚, 韩燕波, 面向服务的计算: 原理和应用, 清华大学出版社, 2006.
- [9] Ana Curi, A Review of Enterprise IT Integration Methods, ITI 2009 31st Int. Conf. on Information Technology Interfaces, June 22-25, 2009.
- [10] Goldberg P R. Survey of Virtual Machine Research[J]. IEEE Computer, 1974, 7(6): 34-35.
- [11] Creasy R J. The Origin of the VM/370 Time-sharing System[J]. IBM Journal of Research and Development, 2006, 25(5): 483-490.
- [12] Smith J E. An Overview of Virtual Machine Architectures[EB/OL]. (2006-04-16). <http://library.intel.com>.
- [13] 魏东, 陈晓江, 房鼎益, 基于 SOA 体系结构的软件开发方法研究, 微电子学与计算机, 2005 年第 22 卷第 6 期. [14] 刘璞, 基于 SOA 的数据交换层关键技术的研究, 硕士论文.
- [15] 曹会敏, 基于 SOA 的服务调用的研究与应用, 硕士论文.
- [16] 鲁松. 计算机虚拟化技术及应用[M]. 北京: 机械工业出版社, 2008.
- [17] Uhlig R, Weiger G, Rodgers D, et al. Intel Virtualization Technology[J]. IEEE Internet Computing, 2005, 38(5): 48-56.
- [18] VMware, Inc.. Live Migration of Virtual Machines Without Service Interruption[EB/OL]. (2008-09-02). <http://www.vmware.com/products/vi/vc/vmotion.html>.
- [19] Clark C, Fraser K, Hand S, et al. Live Migration of Virtual Machines[C]//Proc. of the 2nd ACM/USENIX Symp. on Networked Systems Design and Implementation. Boston, USA: [s. n.], 2005: 273-286.
- [20] Freedman N, Germanow A. IDC's Definition of the Unified Communications Ecosystem[EB/OL]. (2007-01-21).

参考文献

- <http://www.marketresearch.com/IDC-v2477/IDC-Definition-Unified-Communications-Ecosystem-1460171/view-toc/>.
- [21] 丁兆青,董传良,基于 SOA 的分布式应用集成研究,计算机工程,2007. 5.
- [22] 林闯,孔祥震+,周寰,增强计算系统可信赖性:融合虚拟化和 SOA,Journal of Software, Vol.20, No.7, July 2009, pp.1986-2004.
- [23] 王胜娟,江水,企业集成中的企业服务总线技术,计算机工程第 32 卷第 13 期.
- [24] 张俊妍,陈启买,SOAP 协议性能与安全的研究进展计算机技术与发展第 19 卷 第 6 期 2009 年 6 月.
- [25] Csdn.com,ESB 实现 SOA 企业复杂应用,2008. 8.
- [26] Qinwen.Zhang and Qi.Xu, "Intergrated research on quick response system of supply chain based on SOA and ESB,"Journal of Computer Application,vol.29, Oce.2009, pp.2523-2526.
- [27] Yong.Liu, Wei.Yu, "Performance Prediction of SOA Based on ESB,"MICROELECTRONICS&COMPUTER, vol.26, April.2009, pp.110-113.
- [28] 王胜娟,江水,企业集成中的企业服务总线技术,计算机工程第 32 卷第 13 期.
- [29] 张俊妍,陈启买,SOAP 协议性能与安全的研究进展计算机技术与发展第 19 卷 第 6 期 2009 年 6 月.
- [30] Huanqing.Shao,Jianchu.Kang, "Research and Application of Enterprise Seivce Bus," Computer Engineering and Deisign, vol.33,July. 2007,pp. 220-222.
- [31] Chappell D, Enterprise Service Bus 56-70, <http://www.Fiorano.com>.
- [32] Introducing SpiritWave Real-time ESB, <http://www.SpiritSoft.com>.
- [33] ESB Competitive Analysis, <http://www.Fiorano.com>.
- [34] Schmidt M T, 按需应变的运行环境, <http://www.ibm.com>.
- [35]王艳,杨立彪,基于信息技术的旅游服务创新探讨 旅游世界:旅游发展研究 2011.

致谢

时光荏苒，转眼在南开大学研究生的学习生涯已经接近尾声。

在此论文完成之际，我首先要衷心感谢我的导师。本课题项目在研究和实践过程都得到导师的悉心指导。导师为我指出研究和实践的问题，帮助我滤清问题的所在；指导实践的思路。导师一丝不苟的治学态度、渊博的学识和丰富扎实的实践经验以及一针见血的建议都让我受益匪浅。

我还要感谢公司的项目经理，这篇论文的每个设计细节和实践都离不开他的指导。而他宽容的态度和面对问题时的坦诚，帮助我能够很快的融入项目。在相互交流，相互帮助的环境中，我学到了很多。

我要感谢异构数据平台设计研究的同事们。在工作生活中，教会我很多思考问题的方式和实践方法；这不但在生活上，更在学习研究和实践上，给予了我很大的启发和帮助。

感谢我的父母，无论面对怎样的情况，他们都是我坚实的后盾。最后，感谢评阅、评议论文和答辩委员会的各位专家学者在百忙的工作中能给予指导。

个人简历 在学期间发表的学术论文与研究成果

个人简历: