

独 创 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含未获得_____（注：如没有其他需要特别声明的，本栏可空）或其他教育机构的学位或证书使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：郝俊良

签字日期：2007年5月30日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权学校可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。（保密的学位论文在解密后适用本授权书）

学位论文作者签名：郝俊良

导师签字：丁志红

签字日期：2007年5月30日

签字日期：2007年5月30日

学位论文作者毕业后去向：

工作单位：

电话：

通讯地址：

邮编：

基于 ALE 的 RFID 中间件的设计与实现

摘 要

RFID（无限射频识别）是一种新型的识别技术，被称为 2005 年十大热门技术之一，其应用涉及物流、航空、畜牧业、移动商务、票证管理、生产线自动化等各个行业，应用范围非常广泛。针对 RFID 的应用发展起来的 RFID 中间件是 RFID 的关键技术之一，制约着 RFID 的普及和推广，得到了业界的广泛重视。本文从 RFID 的应用切入，从实际应用中提取出 RFID 中间件需要具备的功能。提出了一套基于 ALE 的 RFID 中间件解决方案，并在实际项目中验证其可行性。

论文首先描述了 RFID 的发展过程，探讨 RFID 中间件的研究现状，分析其在国内外的发展趋势；本文从 RFID 入手，描述 RFID 的工作原理，分析 RFID 的应用特点和关键技术，并对 RFID 的重要应用 EPC 进行了探讨，从实际应用中提取出 RFID 中间件的功能需求，得到本文的开发目标。在明确了开发目标后，本文提出了基于 ALE 的 RFID 中间件的解决方案，ALE 是 EPC global 颁布的应用级别事件（Application Level Event）规范，针对 RFID 中间件和 EPCIS 捕获应用之间，定义了 RFID 事件过滤和采集接口。基于 ALE 设计 RFID 中间件符合国际规范，便于扩展和与其它软件衔接。

本文提出的 RFID 中间件分四部分完成中间件的功能：硬件管理器、事件处理器、控制中心和网关。四个部分各司其职又交互合作，硬件管理器主要负责管理硬件设备，事件处理器则是连接 RFID 和企业信息系统的纽带，控制中心负责与操作员的交互，操作员通过控制中心向 RFID 中间件传达命令，网关则是联系其他三部分的通道，硬件管理器，事件处理器和控制中心通过网关进行交互。

本文以 J2EE 作为开发平台，充分利用 J2EE 高可用性、高可靠性、利于扩展的特性及其对 EJB, Servlet, JSP, XML 等技术的全面支持。采用 B/S 架构，操作员只需用浏览器即可进行所有的管理工作，不需要安装任何的客户端软件。用 SOAP 传输协议，把成熟的基于 HTTP 的 WEB 技术与 XML 的灵活性和可扩展性组合在一起，使得系统能够无缝地进行通信和共享数据。实现不同的系统之间能够用“软件-软件对话”的方式相互调用，打破了软件应用、网站和各种设备之间的格格不入的状态，实现“基于 WEB 无缝集成”的目标。

关键词：RFID，无线射频识别，中间件，EPC，事件处理器

The Design and Implement of RFID Middleware Basing On ALE

Abstract

RFID is a new identification technology. It's called one of the hottest ten technologies of 2005. It's applied in many fields, such as transportation, aviation, stock raising, mobile business, the automation of product line and so on. The RFID middleware, which aims at promoting the development of RFID, is one of the key technologies of RFID. It influences the prevalence of RFID and attracts a lot of attention. This thesis begins from the application of RFID, from practical application, we know what a RFID middleware needs to do and, forward a kind of RFID middleware and validate it in the practice.

At first, the thesis describes the process of RFID'S development, discusses the research status of RFID middleware and analyses the trend of it's development domestically and abroad. Then describes the principle of RFID, analyses the characters of the application and the key technologies, and discusses one of RFID'S main application EPC. From the practical application, we know what a RFID middleware has to do and what we should do to develop a practical RFID middleware. After we know the goal of RFID middleware, we put forward a RFID middleware basing on ALE. ALE, Application Level Event, is a specification published by EPC global. It focuses on the RFID middleware and the application of EPCIS and defines the interfaces of event filtration and collection. A RFID middleware which bases on ALE is easy to upgrade and connect to other software.

The RFID middleware we put forward consists of four parts: Reader Server, ALE, Control Center and Gateway. The four parts is separate but work together. Reader Server sees to manage the hardware equipment; ALE connects RFID and the information system of enterprise; Control Center's duty is to communicate with the operator. The operator sends orders to RFID middleware by Control Center. Gateway is the joint of the three other parts, they communicate with each other by Gateway.

The thesis uses J2EE as the development platform, making full use of the characters of J2EE, such as reliability, high usability, easy to update and it's support of EJB, JSP, XML. The RFID middleware adopts B/S architecture. Thanks to B/S, all the operator needs is the browser to do all the management work without installing any other client software.

The thesis uses SOAP as the transport protocol and combines the

agility of XML and the mature WEB technology which bases on HTTP so that the system can share data and communicate freely. This mode of transport realizes different systems communicating with each other by “software to software” , gets rid of the separation of software, web and all kinds of equipment.

Key words: RFID, Radio Frequency Identification, Middleware, EPC, Application Level Event

第一章 绪论

RFID(无线射频识别)是一种非接触式的自动识别技术,它通过射频信号自动识别目标对象并获取相关数据,识别工作无须人工干预,作为条形码的无线版本,RFID 技术具有条形码所不具备的防水、防磁、耐高温、使用寿命长、读取距离大、标签上数据可以加密、存储数据容量更大、存储信息更改自如等优点,其应用将给零售、物流等产业带来革命性变化^[1]。目前 RFID 应用已经渗透到了商业、工业、运输业、物流管理、医疗保险、金融和教学等众多领域。IDC 统计,至 2005 年底,RFID 将达到 6.83 亿美元,2008 年达到 20 亿美元。Note 则估计到 2010 年,RFID 相关产业的产值是 3000 亿美元。

1.1 选题背景

RFID 的应用是一个高度复杂的产业化问题,标准和互操作化程度低就难以形成产业规模,没有规模就难以降低成本,对行业应用形成普及和规模覆盖。

RFID 的应用分成三个阶段:

第一阶段——企业在内部采用了 RFID 技术,但应用的目的多半是为验证 RFID 能够被用在企业的某种环境下,强调 RFID 硬件的可用,对 RFID 应用带来的海量信息没有做有针对性的处理和分析;

第二个阶段——企业把 RFID 和现有的业务流程结合在一起。虽然在应用上迈了一大步,但也是把 RFID 作为原有业务系统的一种完善和补充,仅仅是把 RFID 读取的数据整合到传统的业务流程中,新瓶装旧酒——用新技术来适应传统的应用,没有业务流程上的创新,因此谈不上 RFID 应用的 ROI(投资回报率);

第三个阶段是以 RFID 为核心设计业务流程,因为 RFID 应用从本质上不仅是要收集数据并把数据传送到 IT 设备进行处理,更重要的是要把信息实时回传以便做出合理的决定,因此必须定义出新的业务流程。只有在第三阶段,以 RFID 为核心设计业务流程,才能真正体现 RFID 的商用价值。

由此可见,把 RFID 技术作为条码应用的一个补充或是对原有业务系统的完

善都是无法体现出 RFID 的核心价值的,只有当 RFID 通过企业系统中的核心业务流程实现企业应用,利用外部 workflow 来驱动企业系统,才能实现 RFID 价值的最大化,给企业以看得见、摸得着的 ROI,RFID 中间件应运而生^[6]。

1.2 RFID 中间件的发展历程及发展现状

1.2.1 RFID 中间件的发展历程

最先提出 RFID 中间件概念的是美国。美国企业在实施 RFID 项目改造期间,发现最耗时、耗力、复杂度和难度最高的问题是如何保证 RFID 数据正确导入企业的管理系统,为此企业做了大量的工作用于保证 RFID 数据的正确性。经企业和研究机构的多方研究、论证、实验,最终找到了一个比较好的解决方法,这就是 RFID 中间件。因此,RFID 中间件是随着 RFID 的应用应需而生的,从发展趋势来看,可分为图 1 所示的三大类^[4]:

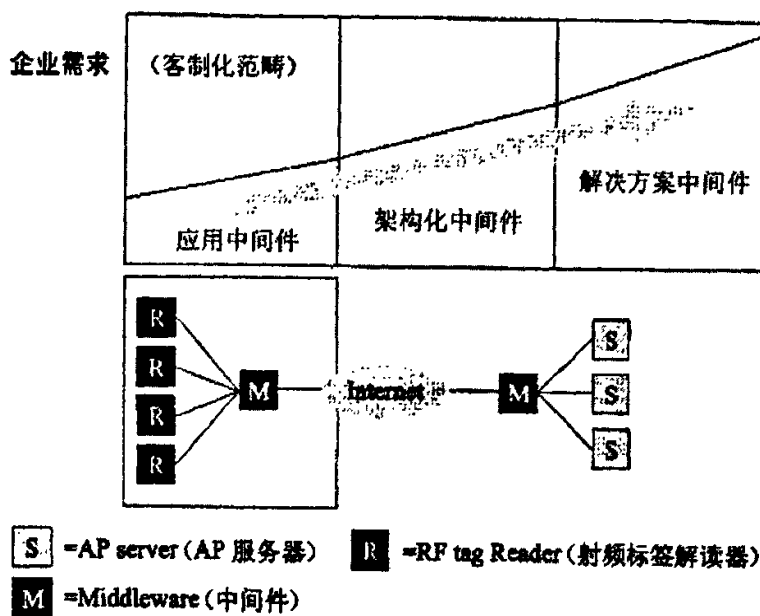


图 1-1 RFID 中间件发展趋势

1、应用中间件发展阶段

RFID 初期的发展多以整合、串接 RFID 解读器为目的,此阶段多为解读器厂商主动提供简单 API,供企业将后端系统与 RFID 解读器串接,例如德州仪

器、EMS、Alien Technology 等^[2]。从整体发展架构来看,此时企业的导入尚须自行花费许多成本去处理前后端系统连接的问题,通常企业在本阶段会透过 Pilot Project(试行项目)方式来评估成本效益与导入的关键议题^[12]。

2、架构化中间件发展阶段

本阶段是 RFID 中间件成长的关键阶段。Wal-Mart 宣布,到 2005 年底所有供应 Wal-Mart 百货公司的商品装箱上,均要求安装有 RFID, Wal-Mart 等国际巨头的推动加上美国国防部的硬性要求,促使各国际大厂持续关注 RFID 相关市场的发展。处在系统架构中间件层的厂商,例如 BEA、Oracle、Sun、IBM、TIBCO 等,都开始在自己原有商品的基础上开发 RFID 中间件;也有些厂商(例如 Oracle、Manhattan Associates 等)与 RFID 解读器厂商合作开发特定的解决方案,以加快进入市场的速度与竞争力。本阶段 RFID 中间件的发展不但已经具备基本的资料搜集、过滤等功能,同时也能满足企业多对多的连接需求,并具备平台的管理与维护功能。

3、解决方案中间件阶段

未来的趋势是,随着 RFID 标签、阅读器与中间件的逐渐成熟,各厂商会针对不同领域提出创新应用的解决方案,Manhattan Associates 公司已经提出“RFID in a Box”的概念,把 RFID 看成一个很盒子,只从盒子抽出一个接口,企业只需通过这个接口利用 RFID 数据,而无需关心盒子内部如何运作,解决了前端 RFID 硬件与后端应用系统的连结问题。该公司与 Alien Technology 公司开展 RFID 硬件端的合作,发展以 Microsoft. Net 平台为基础的中间件,针对该公司 900 家已有的供应链客户群发展一种称为 Supply Chain Execution (SCE) Solution 的解决方案,使原本用 SCE Solution 的企业只需透过“RFID in a Box”,即可在原有应用系统上快速透过 RFID 来加强供应链管理的透明度^[13]。

随着 RFID 的迅速推广,以 RFID 为核心设计业务流程是必然的趋势,因此以 RFID 为中心的解决方案中间件是未来 RFID 中间件的发展趋势。

1.2.2 国外现状

目前国外众多知名厂商都积极投入 RFID 中间件的开发,其中有影响的如 IBM、BEA、SAP、Microsoft, Oracle, Sun 等公司的中间件产品^[10]。

1、BEA 的 RFID 中间件解决方案

BEA 从两个层面实现 RFID 中间件: BEA Web Logic RFID Edge Server 和 BEA Web Logic RFID Enterprise Server。前者直接与硬件设备相连,用于控制读写器按照预定的方式工作,保证不同的设备之间的协调配合。同时它还能按照一定的规则过滤数据,筛除大部分冗余数据,将真正有效的数据传送给后台的信息系统。可提供远程管理、边缘流程管理、WAN 连接层管理等功能。该 Edge Server 虽然叫做服务器,但实际占用空间并不大,甚至可以运行在一些嵌入式设备中,用于远程设备的边缘计算,在工厂、配送中心、零售商店网点等远程数据采集点都可方便得安装和使用。BEA Web Logic RFID Enterprise Server 的物理位置则是介于 Edge Server 和企业后端的应用服务器之间,主要起三个作用:首先,集中管理所有的 Edge Server,通过 Edge Server 兼容各种符合 EPC 标准的 RFID 硬件设备;其次,集中管理各种 RFID 事件,并负责与合作伙伴的信息交换,用户可以围绕这些数据进行业务流程的创新,开发新的应用;最后,它能与各种现有应用系统进行很好的集成,从而产生真正的商业价值。

BEA 参与了国内的主要的 RFID 试验项目,包括上海妇女用品商店基于 RFID 的 CRM 系统,香港的 GS1 的 EPC network 计划,青岛海尔的 RFID 项目,上海百联在物流领域的 RFID 项目等代表性项目。BEA 在流程以及架构上的做法,被业界普遍认为是在为 RFID 中间件产品“定调”。

2、SAP 的 RFID 中间件解决方案

SAP 在韩国成立射频识别标签(RFID)研发中心,拓展 RFID 技术发商务软件开发之后,又携手 Sybase 成立全球联合联盟,整合两家公司的 RFID 中间件。双方的合作层面主要围绕 Sybase 的射频中间件。Sybase RFID Anywhere 2.0 安全中间件通过了 SAP 认证,可与 SAP 自动识别基层设施中间件协同工作,SAP 的自动识别基层设施的功能是将 RFID 标签数据整合进入企业应用程序。在此之前,SAP 还曾与英特尔合作共同开发 RFID 数据直接导入系统,为英特尔的 RFID 设备提供无缝集成方案,也为读卡机提供更多的智能特性。

自 Sybase 收购生产远程管理软件公司 Excellent,并使用其技术开发 RFID 中间件产品后,Sybase 开始大量生产 RFID 电子数据采集设备及软件,包括以为 RFID 读取器和条形码扫描器以及打印机提供一个接口和系统管理工

具。该软件还可以自动将 RFID 数据载入 SQL 数据库,并提供专门的 RFID 数据库模型以存储其他与业务相关数据。Sybase 公司 RFID 主管 Chris Foley 表示,整套 RFID 软件包可以将标签阅读器收集的数据集成到公司的 IT 系统中以供分析。这一产品的目标是那些处于混合结构环境的公司,其中包括对将 RFID 数据整合入 SAP、Oracle 和其他 ERP 系统提供支持。

3、Sun 的 RFID 应用整体解决方案^[3]

Sun 公司提出了 RFID 应用整体解决方案,不论是同设备打交道的 RFID 边缘中间件产品,还是作为搭建企业应用主干的业务集成系统,Sun 公司都能够提供全面、领先的配套产品,并且从架构角度提供了基于业界标准的成熟解决方案。Sun 公司的 RFID 中间件 Sun Java System RFID Software 由两部分组成:Java System RFID Event Manager 和 Java System RFID Information Server。前者用来处理来自一个或多个 RFID Reader、Sensor 的数据流,包括过滤和继承。它的主要功能模块包括:

1) Device Adapter: 适配器允许来自不同厂商的设备能够同 RFID Event Manager 通讯和交互。

2) Filter: 过滤器能够过滤 RFID 设备所提供的冗余数据,还可以用来实施小规模的数据处理和业务逻辑。

3) Connectors: RFID Event Manager 中的 Connector 模块可以将相关信息发送到文件系统、JMS 队列,XML 语言,HTTP,SOAP 消息,从而可以将 RFID 或者是非 RFID 的相关事件数据通知给外部系统。

4) Enterprise Gateway: 该模块可以作为连接企业应用程序的公共接口

Java System RFID Information Server 提供底层的 RFID 数据同高层业务应用的连接通道,可以获取的数据包括:

- 1) 通过 RFID Event Manager 所获取的来自 reader 或 sensor 的数据。
- 2) 标签标识的物品的特征数据,例如:制造日期、重量、失效日期等。
- 3) 产品目录信息。

另外,Sun 公司基于 Java ME,提供了面向 RFID 设备的、支持 EPC Global 事件处理器(Application Level Events)规范的嵌入式软件包:Sun Java System RFID Software for Java ME Devices,此软件包使得支持 Java ME 技术的 RFID 设

备能直接的、智能的处理、过滤设备所产生的相关数据，大幅度减少了 RFID 设备向网络环境发送的数据流量，并以集中的方式来管理大量的分布式 RFID 设备的能力，大大提升了 RFID 设备的智能化程度和管理功能。Sun 公司还提供了便捷的开发方案 RFID Software Toolkit, 用户可以通过这个开发包迅速开发面向不同 RFID 终端设备的 adaptor。

Sun 公司的 Java System RFID 软件架构充分考虑了大规模的、企业级的 RFID 部署环境。这套软件架构不仅适用于只有几个 reader 的单个场所的小规模应用场景，也同样适用于具备大量 reader 并且分布于不同场所的跨地域的大规模部署场景。Java System RFID Event Manager 所特有的可靠性，可扩展性、可管理性，将为企业大规模的部署基于 EPC global Network 架构的应用模式提供良好的集成模块。同时，在将 RFID 标签数据、事件同现有的业务流程、EIS 系统的集成方面，Java System RFID 软件也提供了良好的支持手段。

综观国外软件产品，其做法多是将 RFID 技术纳入其现有中间件产品的软件体系中，RFID 作为可选子项，如 IBM 将 RFID 纳入 Web Sphere 架构，SAP 在 Net Weaver 中增加 RFID 功能。这种在现有产品基础上开发 RFID 模块的方式，优点是开发工作量小、技术成熟度高、而且产品集成性好；缺点是使得 RFID 中间件产品变得庞大、推出“套餐”价格高，并且对系统的依赖性比较大，不利于中小企业低成本轻量级应用。

1.2.3 国内现状

RFID 技术进入中国的时间比较短，各方面的工作还处于起始阶段。虽然我国政府在国家十一五规划和 863 计划中，对 RFID 应用提供了政策、项目和资金的支持，并且 RFID 在国内的发展也较为迅速，但与国际技术的发展相比，在很多方面还存在明显的差距。目前我国做中间件的企业很多，但专门开发 RFID 中间件的企业却很少。

国内在 RFID 中间件和公共服务方面已经开展了一些工作。依托国家 863 计划“无线射频关键技术研究”课题，中科院自动化所开发了 RFID 公共服务体系基础架构软件和血液、食品、药品可追溯管理中间件。华中科技大学开发了支持多通信平台的 RFID 中间件产品 Smart，上海交通大学开发了面向

商业物流的数据管理与集成中间件平台。

虽然中国国内目前已经有了一些初具规模的 RFID 中间件产品，但大多没有在企业进行实际应用测试，与国外的 RFID 中间件产品相比，还处于实验室阶段。与国外经历了很长时间企业实际测试的 RFID 中间件产品相比，还有较大的距离。国内的相关厂家应尽快完成 RFID 中间件产品的企业测试，完善 RFID 中间件的相关功能，为国内中小企业的 RFID 项目实施提供方便、实用、低成本的 RFID 中间件解决方案。

如果国内的企业能够赶在企业开始大规模实施 RFID 项目之前，开发出完善、成熟、可靠的 RFID 中间件产品，加上国内的天时、地利、人和、成本优势，占据中国国内的 RFID 中间件市场是完全有可能的。

通过对比国内外 RFID 中间件的实际情况，不难发现，国外的 RFID 中间件产品发展的时间并不比中国 RFID 中间件早很多；只要中国软件企业奋起直追，依托国内较低的成本优势、众多优秀的技术人员，在短时间内完全有可能开发出与国外的同类产品相匹敌的 RFID 中间件产品。

1.3 课题来源与主要内容

中国海洋大学信息工程中心从 2003 年就一直密切关注 RFID 的发展，做了很多 RFID 的应用研究，这其中包括：

- 1、开展了前置式 RFID 读写器一体机的研究与开发，把数据的筛选过滤等功能提前到读写器端，直接把有用数据发送到后台系统。

- 2、与青岛中科英泰科技有限公司合作开展了“基于嵌入式系统的 RFID 读写器软件”的开发和 HR100 型手持式和 SR200 型台式 RFID 读写器的研制

- 3、开展了基于 RFID 技术的停车场智能化管理系统和电子门票管理系统

- 4、在海尔特种冰箱生产线上进行了 RFID 的现场试点应用。应用中采用先试验、后推广，从前工序向后工序逐步推进、稳扎稳打，从条码技术和 RFID 双轨运行、平稳过渡到 RFID 的策略；避免因技术和管理、流程上的波动影响现场生产。

在以上这些项目中，RFID 的读写设备能得到大量数据，企业本身一般也会有信息系统，像 SCM、ERP、CRM 等，但是如何把两者有效的结合起来，发

挥 RFID 数据的最大作用，一直是制约 RFID 应用的瓶颈。

本项目的目标就是根据实际需要开发一套 RFID 中间件，连接 RFID 的海量数据和企业的本身的信息系统。并将我们的解决方案应用到以上项目中验证其可行性。

本论文共分为五部分：第二部分重点分析 RFID 中间件的相关知识和应用背景，第三部分提出中间件的开发目标，在明确开发目标的基础上提出解决方案的框架结构，并讨论其主要实现技术，第四部分论述中间件的具体实现。第五部分对 RFID 中间件的前景进行了展望。

第二章 RFID 中间件的应用背景

本论文的目标是开发 RFID 中间件，目的是更有效的利用 RFID，因此在开发之前必须详细分析 RFID，了解其工作原理及应用特点，分析其应用背景，才能设计出合理的解决方案。

本章在分析 RFID 及相关应用的基础上，提出 RFID 中间件的开发目标，确定 RFID 中间件的整体开发方案。

2.1 RFID 的工作原理及应用特点

2.1.1 RFID 的工作原理

RFID 是 Radio Frequency Identification 的缩写，即射频识别^[8]。最基本的 RFID 系统由三部分组成：

1、标签(Tag)

标签由耦合元件及芯片组成，每个标签具有唯一的电子编码，附着在物体上标识目标对象；RFID 标签分为被动标签(passive tags)和主动标签(active tags)两种。主动标签自身带有电池供电，读些距离较短同时体积较大，与被动标签相比成本更高，也称为有源标签。被动标签有阅读器产生的磁场中获得工作所需的能量，成本较低并具有很长的使用寿命，比主动标签更小也更轻，读写距离则较近，也称为无源标签。

2、阅读器(Reader)

读取(写入)标签信息的设备,可设计为手持式或固定式。由发送器,接收仪、控制模块和收发器组成。收发器和控制计算机或可编程逻辑控制器(PLC)连接从而实现它的沟通功能。解读器也有天线接收和传输信息。

3、天线(Antenna)

在标签和读取器间传递射频信号。

标签进入磁场后,接收解读器发出的射频信号,凭借感应电流所获得的能量发送出存储在芯片中的产品信息(Passive Tag, 无源标签或被动标签),或者主动发送某一频率的信号(Active Tag, 有源标签或主动标签);解读器读取信息并解码后,送至中央信息系统进行有关数据处理。其原理如图 2-1

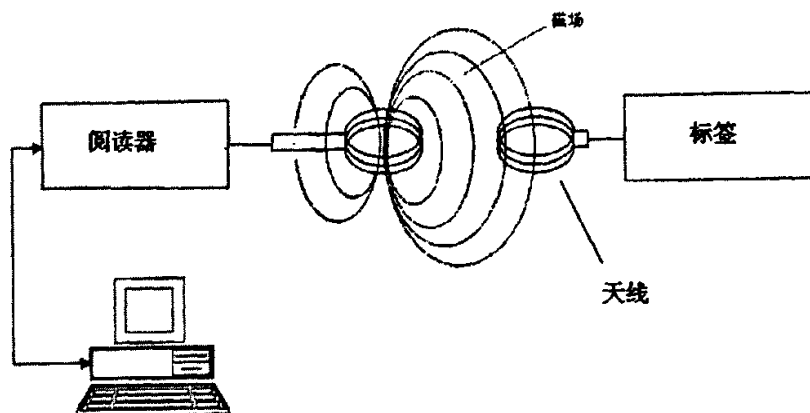


图 2-1 RFID 原理图

RFID 是一种非接触式的自动识别技术,它通过射频信号自动识别目标对象并获取相关数据,识别工作无须人工干预,可工作于各种恶劣环境。可识别高速运动物体并可同时识别多个标签,操作快捷方便,短距离产品不怕油渍、灰尘污染等恶劣的环境,可在这样的环境中替代条码,例如用在工厂的流水线上跟踪物体。长距产品介绍多用于交通上,识别距离可达几十米,如自动收费或识别车辆身份等。作为传统条形码的无线版本,RFID 具有条形码不可比拟的诸多优点:

快速扫描,同时处理多个标签:条形码一次只能有一个条形码受到扫描;
RFID 辨识器可同时辨识读取数个 RFID 标签;

无需光源,可透过外部材料读取数据:在被覆盖的情况下,RFID 能够穿透纸

张、木材和塑料等非金属或非透明的材质，并能够进行穿透性通信。而条形码扫描机必须在近距离而且没有物体阻挡的情况下，才可以辨读条形码；

可重复使用，目前的条形码印刷上去之后就无法更改，RFID 标签则可以重复地新增、修改、删除 RFID 卷标内储存的数据，方便信息的更新；

数据存储容量大，一维条形码的容量是 50Bytes，二维条形码最大的容量可储存 2 至 3000 字符，RFID 最大的容量则有数 MegaBytes。随着记忆载体的发展，数据容量也有不断扩大的趋势。未来物品所需携带的资料量会越来越大，对卷标所能扩充容量的需求也相应增加^[31]；

可读写，可以写入及存取数据，写入时间相比打印条形码更少，标签的内容可以动态改变；

安全性，由于 RFID 承载的是电子式信息，其数据内容可经由密码保护，使其内容不易被伪造及变造。

据全球著名的市场调研公司 Allied Business World 报告显示，2002 年全球 RFID 市场规模是 11 亿美元，2005 年达到 30 亿美金，预计 2010 年可达 70 亿美金，复合年平均增长率为 26%。

2.1.2 RFID 涉及的关键问题

虽然 RFID 技术早在 1948 年就发明出来了^[14]，但其真正受到重视却是在 2003 年。针对 RFID 的应用还不是很成熟，没有形成大规模应用，其应用被一些相关技术和标准所影响或制约。

1、频段

RFID 主要利用无线电波传递信息，在实际应用中主要有三个频段：低频（125KHz）、高频（13.56MHz）以及超高频（900MHz、2.45GHz）。不同的频率段有不同的特点，应用的领域也不一样。低频标签比超高频标签便宜，节省能量，穿透非金属物体力强，他们最适合用于含水成分较高的物体，例如水果等。超高频作用范围广，传送数据速度快，但是比较耗能，穿透力较弱，作业区域不能有太多干扰。

低频 RFID(125KHz)主要用于动物管理，如追踪牛群等动物成长过程中防疫等记录。此外，低频 RFID 由于具备频率低、不易受干扰等特色，因此也适用

于煤气钢瓶的管理,因为钢瓶为金属材质,采用低频 RFID 能避免干扰的问题。高频 RFID(13.56MHz)目前应用颇为普及,门禁控制、交通运输、资产管理(图书、服装等)是典型的应用。超高频 RFID(900MHz、2.45GHz)通常读写距离较远,适用于仓储物流与供应链管理。ST 将 RFID 的应用按照传输距离分成:短距离(0cm~20cm)、长距离(1m 以下)、超长距离(超过 1m)三种应用范围。在短距离与长距离的应用中,ST 采用 13.56MHz 频段,短距离主要应用为身份识别、公交系统范围、防伪,长距离 RFID 产品符合 ISO15693 标准,可应用于交通、门禁。

从市场大小来看,现阶段低频 RFID 使用量最大、应用最为成熟,约占整体市场的 56%,其次为高频 RFID,超高频 RFID 居末。然而由于超高频 RFID 在消费市场的应用备受看好,因此预计所占比重终将超越低频与高频 RFID。

2、标准

任何技术要达到规模应用都必须要有统一的标准,RFID 还未形成统一的全球化标准,市场为多种标准并存的局面。RFID 的标准涉及数据格式、电子标签、读写器、RFID 中间件、空间接口、一致性测试等方面。RFID 系统主要由数据采集和后台数据库网络应用系统两大部分组成。目前已经发布或者是正在制定中的标准主要是与数据采集相关的,其中包括电子标签与读写器之间的空气接口、读写器与计算机之间的数据交换协议、RFID 标签与读写器的性能和一致性测试规范、以及 RFID 标签的数据内容编码标准等。后台数据库网络应用系统目前并没有形成正式的国际标准,只有少数产业联盟制定了一些规范,现阶段还在不断演变中。

目前 RFID 存在两个技术标准阵营,一个是总部设在美国麻省理工学院的 Auto-ID Center,另一个是日本的 Ubiquitous ID Center (UID)。前者的领导组织是美国的 EPC (电子产品代码) 环球协会,旗下有 Wal mart 集团、英国 Tesco 等 100 多家欧美的零售流通企业,同时有 IBM、微软、飞利浦、Auto-ID Lab 等公司提供技术研究支持。后者主要由日系厂商组成,有日本电子厂商、信息企业和印刷公司等,总计达 352 家。目前 EPC global 已经制定或正在制定的标准如表 2-1:

活动种类	相关标准	状态
EPC 物理对象交换	UHF Class0 Gen1 射频协议	已被 Class1 Gen2 超高频标准取代
	UHF Class1 Gen1 射频协议	已被 Class1 Gen2 超高频标准取代
	高频 Class1 Gen1 标签协议	待批准
	Class1 Gen2 超高频空中接口协议	已批准
	EPC 标签数据标准	已批准
	900MHZ Class0 射频识别标签规范	待批准
	13.56MHZ ISM 频段 Class1 射频识别标签接口规范	已批准
	860-930MHZ 频段 Class1 射频识别标签与逻辑通信接口规范	已批准
EPC 基础设施	EPCglobal 体系框架	待批准
	应用水平事件规范	已批准
	读写器协议	制定中
	读写器管理规范	制定中
	标签数据解析协议	制定中
EPC 数据交换	EPCIS 数据规范	制定中
	EPCIS 查询接口规范	制定中
	对象名解析业务规范	已批准
	EPCIS 数据获取接口规范	制定中
	EPCIS 发现协议	待制定
	用户认可协议	待制定

表 2-1 EPC Global 体系框架的标准制定情况

关于标准的另一个问题是，目前还没有正式的关于 RFID 产品（包括各个频段）的国际标准，目前各个厂家推出 RFID 产品互不兼容，造成了 RFID 产品在不同市场和应用上的混乱和不兼容。这势必对未来的 RFID 产品互通和发展造成了阻碍。

通过以上分析总结出 RFID 技术有以下特点：

1) 分布性：每个商品上都贴有标签，商品的分散导致标签的分散，大规模

应用导致 RFID 设备的分布性和多样性, 具有物理上的分布等特点:

2) 异构性: 各种类型的读写器, 协议在分布式应用环境中存在;

3) 海量数据的冗余性: 因为 RFID 是实时采集数据的, 因此必然会得到大量的数据, 在大规模应用环境中, 容易造成数据重复读入

4) 数据传输的不可靠性: 在无线应用环境中, 信号通过天线传输给读写器, 再通过网络传输到各个应用系统, 其中前者容易有数据丢帧, 丢包等现象, 需要提高数据传输的质量和可靠性;

6) 数据的不完整性: 由于数据传输的可靠性和数据冗余性易造成数据的不完整性和不一致性。

一方面是大量 RFID 硬件提供的海量、冗余数据, 一方面是企业对 RFID 数据精简、可靠, 满足本企业现有业务系统的要求, RFID 硬件提供的数据显然不能满足企业的需求。事实上, 这个问题的本质是用户应用系统与硬件接口的问题。在 RFID 应用中, 通透性是整个应用的关键, 正确抓取数据、确保数据读取的可靠性、以及有效地将数据传送到后端系统都是必须考虑的问题。从技术上解决这些问题的最好办法是采用中间件技术, 作为硬件设备和应用的中枢产品, 根据业务系统的特点, 进行信息的整理、分类、并根据业务流程进行业务系统的配置, 以保证能及时地向企业应用系统提供业务所需的数据^[1]。

RFID 中间件与 RFID 的读写器相连。读写器不停地从标签中采集 RFID 标签数据, 并向 RFID 中间件传输。每次识读 RFID 中间件都要保存一些信息, 例如: 标签的编码数据、扫描标签的读写器的唯一标识码、识读时间、以及其它相关的一些信息, 如读写器地理位置等。RFID 中间件是程序模块的集成, 程序模块通过两个接口与外界交互, 读写器通过连个接口与外界交互, 读写器接口和应用程序接口。其中读写器接口提供与标签读写器, 尤其是 RFID 读写器的连接方法。应用程序接口使 RFID 中间件和外部应用程序连接, 这些应用程序通常是现在的企业正在使用的应用程序, 也有可能新的特定的 RFID 应用程序甚至是其他 RFID 中间件。除了 RFID 中间件定义的两个外部接口(读写器接口和应用程序接口)外, 程序模块之间用它们自己定义的 API 函数交互。

2.2 RFID 在 EPC 的应用

2.2.1 什么是 EPC

EPC 系统是 RFID 技术最重要的应用领域之一^[3]。EPC(电子产品代码)是用来在供应链中唯一标识商品的编码,这个编码存储在 RFID 新片中,一件商品对应一个电子产品代码。电子产品代码能够辨别具体对象的生产者,产品、定义,序列号。它除了具有全球标准代码能辨识物体的功能外,还可以通过电子产品代码网络提供关于产品的附加信息,例如产地,产品历史等,这些数据对于在供给链中特定产品的历史追踪具有关键的作用^[15]。

EPC 网络是一项能够实现供应链中的商品快速自动识别以及信息共享的技术。商品数据被储存在互联网或其他网络上,只要使用标准的技术就可以进入数据系统,就像进入互联网一样,使供应链中商品信息真实可见,指导企业利用这些数据做出正确决策。总的来说 EPC 就像是一把钥匙,用以解开 EPC 网络上相关产品信息这把锁。

采用 RFID 最大的好处是可以对企业的供应链进行高效管理,以有效地降低成本。因此对于供应链管理应用而言,射频技术是一项非常适合的技术。EPC 产品电子代码及 EPC 系统的出现,使 RFID 技术向跨地区、跨国界物品识别与跟踪领域的应用迈出了划时代的一步。

成熟的 RFID 技术应用于新生的 EPC 系统^[16],将极大拓展 RFID 技术的应用领域,给 RFID 技术特别是 RFID 标签市场带来迅猛增长,随着零售巨擎沃尔玛要求其供应商使用 EPC 射频标签的期限迫近,EPC 给 RFID 世界带来的商机已逐渐显现,同时,随着 2004 年第二代射频标签全球标准的出台,RFID 技术与市场的发展将更加规范有序,EPC 系统的推广与应用将真正步入快车道。

2.2.2 EPC 的结构

EPC 网络由五个基本要素组成:EPC 产品电子代码、识别系统、对象名解析服务(ONS)、中间件以及 EPC 信息服务 EPCIS。

1、EPC 编码

EPC 编码是新一代的与 EAN·U 控制中心编码兼容的编码标准,与现行的

GTIN 相结合 i, 可以在 EPC 网络中兼容 EAN • U 控制中心系统。EPC 编码是由四个部分组成的一串数字, 依次为版本号, 域名管理者, 对象分类, 序列号。编码长度分为 64 位, 96 位和 256 位三种。可以为物理世界的每个对象提供唯一标识。其编码的分配由 EPCglobal 和各国的 EPC 管理机构分段管理, 共同维护。

2、识别系统

识别系统包括 RFID 标签和阅读器两个部分。RFID 标签是一种由一块微芯片和底层天线构成的 RFID 设备, EPC 就是存储在 RFID 标签上, 而为了降低标签的成本, 在 EPC 网络中 EPC 也是 RFID 标签上的唯一信息。RFID 标签采用无线射频 (RF) 技术与阅读器传递 EPC 数据, 而阅读器则通过中间件传递信息到 EPC 网络的信息服务层。

3、中间件

中间件是 EPCglobal 网络的神经中枢, 主要任务是数据校对、读写器协调、数据传送、数据存储和任务管理。中间件的核心组成是事件管理器和信息服务器。事件管理器负责采集、过滤读写器收集的 EPC 相关信息^[17], 并转发给其它应用。信息服务器提供事件管理器与企业信息系统之间的集成, 储存事件服务器提交的数据信息, 提供访问接口。

4、对象名解析服务

ONS 是使用户能够通过物品的 EPC 码提出访问请求并找到与之相关的信息的一套服务系统。它的工作原理和采用的技术类似与互联网中的域名解析服务 (DNS)。DNS 的提供的是根据服务器的网址将其定位到互联网中的某一具体地点的服务, 与之相似, ONS 提供一种全球的搜索服务, 将产品的 EPC 码转换成一个或多个互联网统一参考定位器 (Uniform Reference Locato 硬件管理器 URLs), 通过 URLs 就可以定位 EPC 网络中存储 EPC 码对应的产品信息。这些 URLs 所要定位的主要就是企业 EPC IS 的地址, 当然也有可能找到与 EPC 码对应的网站和其他的互联网资源的地址来实现其定位服务。

5、EPC 信息服务

EPC 信息服务为信息服务请求提供相关数据。这些数据包括由 Savant 系统采集的标签数据 (例如, 为了实现物品跟踪而需要的产品序列号粒度级别的数

据); 实物层的数据, 如: 生产日期、过期日期等等; 物品类别层次数据, 像产品种类信息。为了响应信息服务请求, EPC 信息服务需要从企业内部提取一系列相关的数据并将其转化为 PML 格式。同时, EPC 信息服务还要对不同的信息请求作不同的数据处理, 以满足用户的需要。

EPC 信息服务就像是整个 EPC 网络运行的大脑, 它处理着企业核心的信息服务需求。EPC 网络中的每个企业都管理、控制着自己的 EPC 信息服务, 决定授给谁相应的权限以获取不同的信息服务。

读写器读出的 EPC 只是一个信息参考 (指针), 由这个信息参考从 INTERNET 找到 IP 地址并获取该地址中存放的相关的物品信息, 并采用分布式的中间件处理由读写器读取的一连串 EPC 信息。由于在标签上只有一个 EPC 代码, 计算机需要知道与该 EPC 匹配的其它信息, 这就需要 ONS 来提供一种自动化的网络数据库服务, EPC 中间件将 EPC 代码传给 ONS, ONS 指示 EPC 中间件到一个保存着产品文件的服务器 (EPC IS) 查找, 该文件可由 EPC 中间件复制, 因而文件中的产品信息就能传到供应链上, EPC 系统的工作流程如图 2-2 所示。

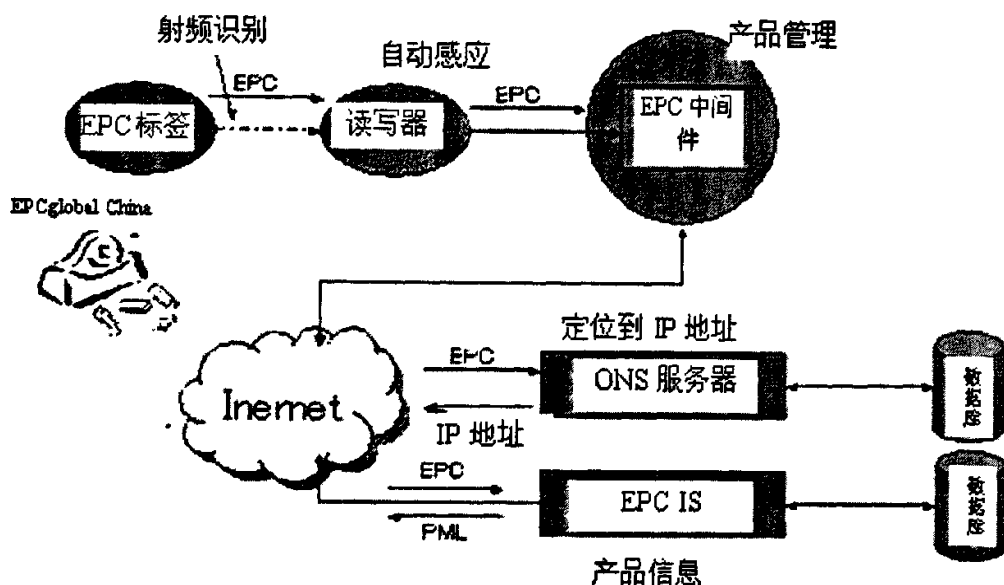


图 2-2 EPC 系统工作流程示意图

2.2.3 RFID 与 EPC

EPC 的目标是建立一个“物联网”，所谓物联网是在计算机互联网的基础上，利用 RFID、无线数据通信等技术，构造一个覆盖世界上万事万物的“Internet of Things”。在这个网络中，物品(商品)能够彼此进行“交流”，而无需人的干预。其实质是利用射频自动识别(RFID)技术，通过计算机互联网实现物品(商品)的自动识别和信息的互联与共享。而 RFID，正是能够让物品“开口说话”的一种技术^[18]。在“物联网”的构想中，RFID 标签中存储着规范而具有互用性的信息，通过无线数据通信网络把它们自动采集到中央信息系统，实现物品(商品)的识别，进而通过开放性的计算机网络实现信息交换和共享，实现对物品的“透明”管理。

RFID 中间件是用来加工和处理来自读写器的所有信息和事件流的软件，是连接读写器和企业应用的纽带。它要对标签数据进行过滤、分组和计数，以减少发往信息网络系统的数据量并防止错误识读、漏读和多读信息。RFID 中间件是物联网的神经系统，是一种企业通用的管理 RFID 标签数据的架构。它可以安装在商店、本地配送中心、区域甚至全国数据中心，以实现数据的捕获、监控和传送。这种分布式结构可以简化物联网的管理，提高运行效率。RFID 中间件可以实时采集 RFID 标签数据。在典型应用下，RFID 中间件软件安装在商店、仓库、制造车间，甚至卡车上。RFID 标签数据只有通过它们才能进入系统^[7]。

RFID 中间件与 RFID 的读写器相连。读写器不停地从标签中采集 RFID 标签数据，并向 RFID 中间件传输。每次识读 RFID 中间件都要保存一些信息，例如：标签的编码数据、扫描标签的读写器的唯一标识码、识读时间、以及其它相关的一些信息，如读写器地理位置等^[19]。

RFID 中间件是程序模块的集成，程序模块通过两个接口与外界交互，读写器通过连个接口与外界交互，读写器接口和应用程序接口。其中读写器接口提供与标签读写器，尤其是 RFID 读写器的连接方法。应用程序接口使 RFID 中间件和外部应用程序连接，这些应用程序通常是现在的企业正在使用的应用程序，也有可能新的特定的 RFID 应用程度甚至是其他 RFID 中间件。除了 RFID 中间件定义的两个外部接口（读写器接口和应用程序接口）外，程序模块之间用它们自己定义的 API 函数交互。

2.3 RFID 中间件的开发目标

通过对 RFID 技术及其相关应用的分析,对 RFID 中间件的概念和功能有了大体的了解,总结一下,一个典型的 RFID 中间件基本上需要包括以下功能,也是本文所要开发的 RFID 中间件所要达到的目标:

1、提供与多种 RFID 读写器兼容设备操作接口

屏蔽硬件差异性,兼容主流电子标签编码标准;读写 RFID 读写器种类变化或增加时,应用端不需修改,解决了多对多连接的维护复杂性问题。

2、数据过滤和传输

在输入的巨量事件中发现有用的和重要的事件,过滤冗余的、无关的数据,减小事件的数量。并把时间传输到后台应用程序,高效地处理大批量事件数据,减少数据处理和传输中对后台数据库的频繁操作和因存储、查询所带来的数据在网络中的来回传输^[6]。

3、管理 RFID 读写设备

终端使用者能够通过 RFID 中间件接口直接的装配、监视、部署和发送命令至读卡器。例如:使用者能够告诉读卡器何时通过“turn off”来避免无限射频冲突。

4、支持与已有的业务系统集成

提供的一组通用应用程序接口(API),实现从应用程序端到 RFID 读写器的连接。即使存储 RFID 标签数据的数据库软件或后端应用程序增加或改由其它软件取代,应用端不需修改也能处理,解决多对多连接的维护复杂性问题^[9]。

5、支持多应用系统请求 RFID 数据 提供统一管控平台,支持多种网络和通信接入方式。

RFID 中间件在 RFID 应用框架中的位置如图 2-3:

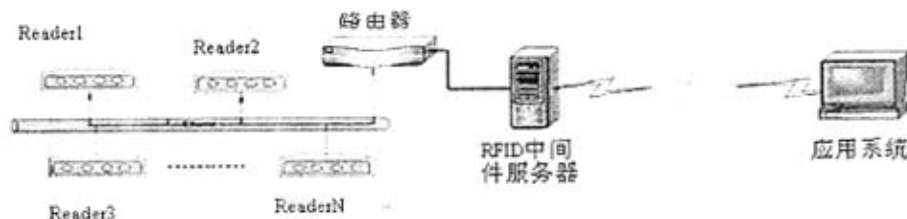


图 2-3 RFID 中间件在 RFID 应用框架中的位置

2.4 RFID 中间件的框架结构

一个先进的、设计合理的 RFID 中间件，不仅要求功能全面，还要求易升级、易扩展，方便与系统的衔接。

在设计当中，为了使各功能模块能独立工作，减少彼此之间的依赖性，采用模块化结构设计，每一个模块完成特定的功能，各模块之间相互联系，使软件具有较高的可移植性和可扩展性^[20]。

根据 RFID 中间件的功能，本文分四部分完成中间件：控制中心，硬件管理器，事件处理器，网关，这四部分相互协作，共同完成 RFID 中间件的功能。

RFID 中间件的结构框图如图 2-6 所示：

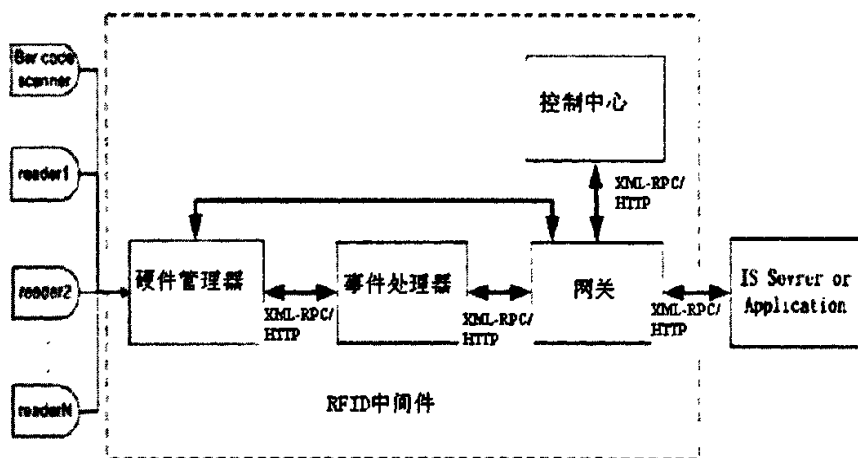


图 2-4 基于 ALE 的 RFID 中间件结构图

1、硬件管理器

硬件管理器主要负责管理硬件设备，控制硬件设备的开启、关闭和配置；对原始数据进行初步处理：过滤、存储、传输；提供统一接口，屏蔽硬件差异，实现硬件无关性；其功能包括：

启动和停止识读设备。用户通过控制中心向硬件管理器发送消息，启动或停止某个或全部识读设备；

保存所有相关识读设备的配置信息。所有识读设备的配置信息保存在硬件管理器中，用户可以通过控制中心进行修改，动态配置识读设备的参数；

向控制中心发送识读设备配置信息。硬件管理器启动后会把识读设备的配置信息发送给控制中心，操作员就可以看到该服务器连接的所有识读设备的信息；

响应事件处理器的命令并做相应处理。事件处理器也需要配置一些识读设备的参数，硬件管理器的一个功能就是响应事件处理器的命令并根据命令修改识读设备的相关参数；

将读取的 EPC 信息经过简单处理发送到事件处理器。硬件管理器会把读到的标签信息根据要求作简单处理后送到事件处理器。

硬件管理器与控制中心和事件处理器的交互如图 2-5 所示：

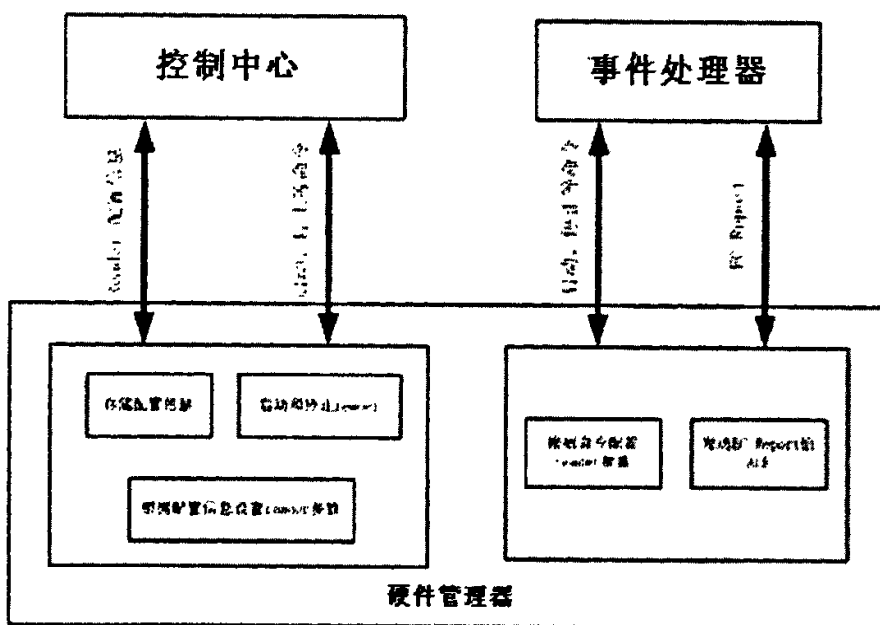


图 2-5 硬件管理器与控制中心和事件管理器的交互图

2、事件处理器

事件处理器主要是对从硬件管理器传送的数据进行合成整理，以及把标签数据封装成标准的数据格式，为上层的应用系统提供服务。其功能包括：

把逻辑识读设备与物理识读设备建立映射。提出逻辑识读设备的概念，主要是便于对物理识读设备的管理，当物理识别设备发生变化时，只需通过控制中心对映射进行修改即可，无需修改上层应用系统；

接收硬件管理器传送的数据。对硬件管理器传送的数据进行合成、过滤；

根据上层应用的定制信息，对服务进行定制。可以设置事件类型、过滤器种类、触发器的类型以及的参数值，还可以对标签数据的进行分组。

3、控制中心

控制中心负责配置管理硬件管理器、事件处理器，以及管理控制物理识读设备。分为系统管理及配置管理两大模块。其中系统管理模块的功能有：

提供系统登录、退出系统，增、删、改、查操作员等操作；

系统登录/退出。操作员只能用正确的帐号才能登录到控制中心平台上来，登录后的界面根据操作员的权限的不同而不同。退出系统时删除与操作员相关的 session 信息，并退出到登录界面来；

增、删、改、查操作员信息。提供完整的管理操作，而且可给不同的操作员赋予不同的权限；

配置管理模块的功能有：

提供配置硬件管理器、硬件识读设备及事件处理器等操作。可设置硬件管理器 及事件处理器的的基本配置信息，以及物理识读器的信息等；

查看系统状态。采用 Applet 及 JSP 两种风格展示系统状态图。操作员可按实际情况选用最合适的方式查看系统状态信息；

控制中心主要是提供操作员控制 RFID 中间件的平台，用户通过控制中心对 RFID 中间件进行配置，操作员与系统的交互活动如图 2-6 所示：

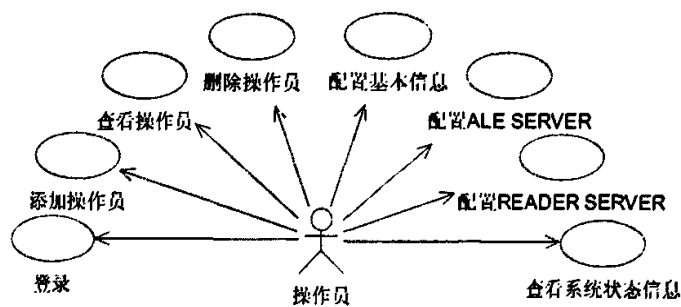


图 2-6 控制中心可执行的操作

4、网关

网关主要实现管理服务 and 数据服务协议 的转换。它是连接中间件各功能模块的枢纽，硬件管理器和事件处理器通过往网关与控制中心交互。

第三章 RFID 中间件主要技术的选取

RFID 的应用涉及物流、航空、畜牧业、移动商务、票证管理、停车场管制、生产线自动化等各个行业，应用范围非常广泛，每个行业都有独特的应用环境，每个企业都有自己原先的信息系统，这就要求 RFID 中间件具有灵活的适应性，能适应各种运行环境，和各种系统兼容。而 RFID 中间件方面的设计和研究还只是刚刚起步的阶段，大部分的研究还都处于试验阶段，存在着或多或少的不足和限制。比如：文献[3]缺乏对 RFID 中间件的支持：功能描述不清，没有统一的事件过滤(filter)规范，不支持高层事件的定义；文献[4]的研究是基于对自动控制领域的经验，而并非完全针对 RFID 系统，有较大的局限性；文献[5]CEP 研究的示范应用主要在网络的入侵检测与安全、企业监控和管理两个方面，还没有 RFID 系统的示范应用。而 Microsoft、IBM、Oracle、Sybase 等厂商所提的 RFID 中间件的设计大多是基于各自的核心产品或技术的应用，有太大的依赖性和较小的扩展性，灵活度不够。一个健壮的 RFID 中间件应当具备以下特点：

遵循开放标准：不管是遵循国内标准或兼容国际标准

业务价值定义：过滤特定行业应用的价值点，进行应用模式的创新，明确业务目标

硬件有效实施：针对不同物理环境和业务目标选择特定技术，辅助以强大的硬件集成能力，提供适应不同环境复杂度的硬件部署和实施

成熟软件架构：兼容现有应用环境，面向未来业务拓展

灵活业务流程：适应不同应用场景下的业务流程定制能力，支持供应链环境下的信息交换 完整业务展现：简化业务操作的同时，为企业决策层提供全面业务透视能力

本文以 I2EE 为平台开发的 RFID 中间件，是基于开放式架构设计的、模块化的，可升级数据处理系统，可以运行在多种操作系统之上，支持大多数主流的应用服务器，具有广泛的适用性。

3.1 J2EE 开发平台

3.1.1 J2EE 体系结构

J2EE (Java 2 Enterprise Edition) 是建立在 Java 2 平台上的企业级应用的解决方案。J2EE 技术的基础便是 Java 2 平台, 不但有 J2SE 平台的所有功能, 同时还提供了对 EJB, Servlet, JSP, XML 等技术的全面支持, 其最终目标是成为一个支持企业级应用开发的体系结构, 简化企业解决方案的开发, 部署和管理等复杂问题。事实上, J2EE 已经成为企业级开发的工业标准和首选平台^[21]。

目前, Java 2 平台有 3 个版本, 它们是适用于小型设备和智能卡的 Java 2 平台 Micro 版 (Java 2 Platform Micro Edition, J2ME)、适用于桌面系统的 Java 2 平台标准版 (Java 2 Platform Standard Edition, J2SE)、适用于创建服务器应用程序和服务的 Java2 平台企业版 (Java 2 Platform Enterprise Edition, J2EE)。

J2EE 是一种利用 Java 2 平台来简化企业解决方案的开发、部署和管理等复杂问题的体系结构, 它的基础是核心 Java 平台或 Java 2 平台的标准版, J2EE 不仅巩固了标准版中的许多优点, 例如“编写一次, 随处运行”的特性、方便存取数据库的 JDBC API、CORBA 技术以及能够在 Internet 应用中保护数据的安全模式等等, 同时还提供了对 EJB (Enterprise JavaBeans)、Java Servlets API、JSP (Java Server Pages) 以及 XML 技术的全面支持。其最终目的就是成为一个能够使企业开发者大幅缩短投放市场时间的体系结构^[28]。

J2EE 体系结构提供中间层集成框架用来满足高可用性、高可靠性以及可扩展性的应用的需求, 并且不需太多费用。通过提供统一的开发平台, J2EE 降低了开发多层应用的费用和复杂性, 同时提供对现有应用程序集成的强有力支持, 完全支持 Enterprise JavaBeans, 有良好的向导支持打包和部署应用, 添加目录支持, 增强了安全机制, 提高了性能。

3.1.2 J2EE 的优势

J2EE 为搭建具有可伸缩性、灵活性、易维护性的商务系统提供了良好的机制, 用 J2EE 作为 RFID 中间件的开发平台具有以下优势:

- 1、保留现存的 IT 资产: 企业引进 RFID, 是在原有企业信息系统的基

础上追加 RFID 解决方案,而不是重新制定全盘方案。这样,一个以渐进的(而不是激进的,全盘否定的)方式建立在已有系统之上的 RFID 中间件是企业所需求的。J2EE 架构可以充分利用用户原有的投资,如一些公司使用的 BEA Tuxedo、IBM CICS, IBM Encina、Inprise VisiBroker 以及 Netscape Application Server 等。这之所以成为可能首先是因为基于 J2EE 平台的产品几乎能够在任何操作系统和硬件配置上运行,现有的操作系统和硬件能被保留继续使用。其次 J2EE 还拥有广泛的业界支持和一些重要的“企业计算”领域供应商的参与。每一个供应商都对现有的客户提供了不用废弃已有投资,进入可移植的 J2EE 领域的升级途径。

2、高效的开发: J2EE 允许企业把一些通用的、很繁琐的服务端任务交给中间件供应商去完成。当企业使用了 RFID 中间件之后,如果需要扩展服务,无需自行开发,可以交给中间件供应商去完成^[29]。这样开发人员可以把精力集中在如何创建商业逻辑上,相应地缩短了开发时间。

3、支持异构环境: J2EE 能够开发部署在异构环境中的可移植程序。基于 J2EE 的应用程序不依赖任何特定操作系统、中间件、硬件。因此设计合理的基于 J2EE 的程序只需开发一次就可部署到各种平台。这在典型的异构企业计算环境中是十分关键的。使用 RFID 技术的企业涉及物流、航空、畜牧业、移动商务和物流等各行各业,企业原有的信息系统非行复杂 J2EE 标准也允许客户订购与 J2EE 兼容的第三方的现成的组件,把他们部署到异构环境中,节省了由自己制订整个方案所需的费用。

4、可伸缩性: 企业必须要选择一种服务器端平台,这种平台应能提供极佳的可伸缩性去满足那些在他们系统上进行商业运作的大批新客户。基于 J2EE 平台的应用程序可被部署到各种操作系统上。例如可被部署到高端 UNIX 与大型机系统,这种系统单机可支持 64 至 256 个处理器。(这是 NT 服务器所望尘莫及的) J2EE 领域的供应商提供了更为广泛的负载平衡策略。能消除系统中的瓶颈,允许多台服务器集成部署。这种部署可达数千个处理器,实现可高度伸缩的系统,满足未来商业应用的需要。

5、稳定的可用性: 一个服务器端平台必须能全天候运转以满足公司客户、合作伙伴的需要。因为 INTERNET 是全球化的、无处不在的,即使在夜间按计

划停机也可能造成严重损失。若是意外停机，那会有灾难性后果。J2EE 部署到可靠的操作环境中，他们支持长期的可用性。一些 J2EE 部署在 WINDOWS 环境中，客户也可选择健壮性能更好的操作系统如 Sun Solaris、IBM OS/390。最健壮的操作系统可达到 99.999% 的可用性或每年只需 5 分钟停机时间。这是实时性很强商业系统理想的选择。

综上所述，用 J2EE 作为 RFID 中间件的开发平台，能适应企业复杂多样的原有信息系统，可进行快速高效的开发，并且具有高度的可移植性和可扩展性。

3.2 B/S 体系架构

在系统的开发平台选定之后，选择合理的架构是很重要的。合理的架构不仅方便用户使用，因为它是用户和 RFID 中间件的接口；而且还提高系统资源利用率。开发基于 J2EE 的中间件既要考虑 J2EE 的特性和需求，又要满足系统可靠性和实时性的要求，使得系统最大程度地发挥其功能。当今世界开发模式技术架构的两大主流技术有两种：C/S 结构和 B/S 结构。

3.2.1 C/S 架构

C/S 结构，即 Client/Server (客户机/服务器) 结构。此结构把数据库内容放在远程的服务器上，而在客户机上安装相应软件。C/S 软件一般采用两层结构，其分布结构如图 3-1 所示。

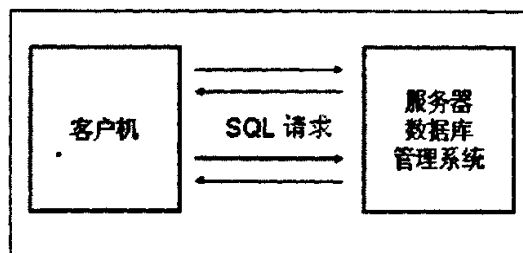


图 3-1 C/S 结构图

C/S 架构由两部分构成：前端是客户机，即用户界面(Client)结合了业务逻辑，接受用户的请求，并向数据库服务提出请求，通常是一个 PC 机；后端是服务

器,即数据管理 (Server) 将数据提交给客户端,客户端将数据进行计算并将结果呈现给用户。还要提供完善的安全保护及对数据的完整性处理等操作,并允许多个客户同时访问同一个数据库。在这种结构中,服务器的硬件必须具有足够的处理能力,这样才能满足各客户的要求。

C/S 结构在技术上很成熟,它的主要特点是交互性强、具有安全的存取模式、网络通信量低、响应速度快、利于处理大量数据。但是该结构的程序是针对性开发,变更不够灵活,维护和管理的难度较大。通常只局限于小型局域网,不利于扩展。并且,由于该结构的每台客户机都需要安装相应的客户端程序,分布功能弱且兼容性差,不能实现快速部署安装和配置,因此缺少通用性,具有较大的局限性。要求具有一定专业水准的技术人员去完成。因此使用 C/S 架构需要高昂的维护成本,投资大。

首先,采用 C/S 架构,要选择适当的数据库平台来实现数据库数据的真正“统一”,使分布于两地的数据同步完全交由数据库系统去管理,但逻辑上两地的操作者要直接访问同一个数据库才能有效实现,有这样一些问题,如果需要建立“实时”的数据同步,就必须在两地间建立实时的通讯连接,保持两地的数据库服务器在线运行,网络管理工作人员既要服务器维护管理,又要对客户端维护和管理,这需要高昂的投资和复杂的技术支持,维护成本很高,维护任务量大。

其次,传统的 C/S 结构的软件需要针对不同的操作系统系统开发不同版本的软件,由于产品的更新换代十分快,代价高和低效率已经不适应工作需要。在 JAVA 这样的跨平台语言出现之后,B/S 架构更是猛烈冲击 C/S,并对其形成威胁和挑战。

C/S 结构分布功能弱且兼容性差,跟 RFID 中间件的需求正好冲突。RFID 的一个特点就是分布式应用,并要求中间件有高度的可扩展性,因此 C/S 结构不适合 RFID 中间件。

3.2.2 B/S 架构

B/S 结构,即 Browser/Server(浏览器/服务器)结构,就是只安装维护一个服务器 (Server),而客户端采用浏览器 (Browse) 运行软件。它是随着 Internet

技术的兴起,对 C/S 结构的一种变化和改进。主要利用了不断成熟的 WWW 浏览器技术,结合多种 Script 语言 (VBScript、JavaScript...) 和 ActiveX 技术,是一种全新的软件系统构造技术。

B/S 三层体系结构采用三层客户/服务器结构,在数据管理层(Server)和用户界面层(Client)增加了一层结构,称为中间件 (Middleware),使整个体系结构成为三层。三层结构是伴随着中间件技术的成熟而兴起的,核心概念是利用中间件将应用分为表示层、业务逻辑层和数据存储层三个不同的处理层次,如图 2 所示。三个层次的划分是从逻辑上分的,具体的物理分法可以有多种组合。中间件作为构造三层结构应用系统的基础平台,提供了以下主要功能:负责客户机与服务器、服务器与服务器间的连接和通信;实现应用与数据库的高效连接;提供一个三层结构应用的开发、运行、部署和管理的平台。这种三层结构在层与层之间相互独立,任何一层的改变不会影响其它层的功能。

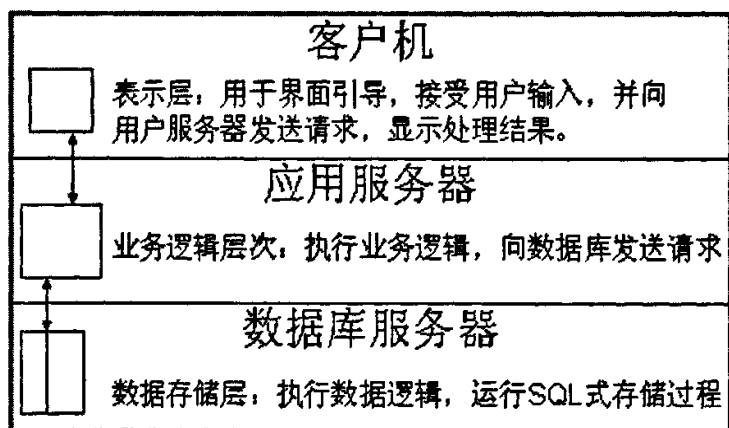


图 3-2 B/S 三层结构体系

在 B/S 体系结构系统中,用户通过浏览器向分布在网络上的许多服务器发出请求,服务器对浏览器的请求进行处理,将用户所需信息返回到浏览器。而其余如数据请求、加工、结果返回以及动态网页生成、对数据库的访问和应用程序的执行等工作全部由 Web Server 完成。随着 Windows 将浏览器技术植入操作系统内部,这种结构已成为当今应用软件的首选体系结构。显然 B/S 结构应用程序相对于传统的 C/S 结构应用程序是一个非常大的进步。B/S 架构具有以下特点:

1、维护和升级方式简单

目前,软件系统的改进和升级越来越频繁,B/S 架构的产品明显体现着更为方便的特性。对一个稍微大一点单位来说,系统管理人员如果需要在几百甚至上千部电脑之间来回奔跑,效率和工作量是可想而知的,但 B/S 架构的软件只需要管理服务器就行了,所有的客户端只是浏览器,根本不需要做任何维护。无论用户的规模有多大,有多少分支机构都不会增加任何维护升级的工作量,所有的操作只需要针对服务器进行;如果是异地,只需要把服务器连接专网即可,实现远程维护、升级和共享。所以客户机越来越“瘦”,而服务器越来越“胖”是将来信息化发展的主流方向。今后,软件升级和维护会越来越容易,而使用起来会越来越简单,这对用户人力、物力、时间、费用的节省是显而易见的,惊人的。因此,维护和升级革命的方式是“瘦”客户机,“胖”服务器^[22]。

2、成本降低,选择更多

大家都知道 windows 在 desktop 电脑上几乎一统天下,浏览器成为了标准配置,但在服务器操作系统上 windows 并不是处于绝对的统治地位。现在的趋势是凡使用 B/S 架构的应用管理软件,只需安装在 Linux 服务器上即可,而且安全性高。所以服务器操作系统的选择是很多的,不管选用哪种操作系统都可以让大部分人使用。windows 作为桌面操作系统的电脑不受影响,这就使的最流行免费的 Linux 操作系统快速发展起来,Linux 除了操作系统是免费的以外,连数据库也是免费的,这种选择非常盛行。比如说很多人每天上“网易”,只要安装了浏览器就可以了,并不需要了解“网易”的服务器用的是什么操作系统,而事实上大部分网站确实没有使用 windows 操作系统,但用户的电脑本身安装的大部分是 windows 操作系统。

B/S 结构的主要特点是分布性强、维护方便、开发简单且共享性强、总体拥有成本低。与 C/S 相比有以下优势:

1、对硬件要求小

C/S 建立在局域网的基础上,通过专门服务器提供连接和数据交换服务。所处理的用户不仅固定,并且处于相同区域,要求拥有相同的操作系统。B/S 建立在广域网的基础上,信息自己管理,有比 C/S 更强的适应范围,一般只要有操作系统和浏览器就行。与操作系统平台关系最小。面向不可知的用户群。

2、软件重用性强

C/S 程序需要不可避免的整体性考虑,构件的重用性不如在 B/S 要求下的构件的重用性好。B/S 对的多重结构,要求构件相对独立的功能,能够相对较好的重用。就如买来的餐桌可以再利用,而不是做在墙上的石头桌子。

3、系统维护简单

系统维护在软件生存周期中是相当重要的。C/S 程序由于整体性,必须整体考察,处理出现的问题。系统升级难,可能是再做一个全新的系统。B/S 则是由构件组成,方便构件的个别更换,实现系统的无缝升级,系统维护开销减到最小,用户从网上自己下载安装就可以实现升级。

4、用户群更广泛

C/S 程序处理的问题要求用户面固定,并且在相同区域,安全性能高,并且与操作系统相关,必须都是相同的系统。B/S 则是建立在广域网上的,面向分散地域上的不同用户群,这是 C/S 无法作到的,并且与操作系统平台关系最小。这时的客户机成为瘦客户机,而服务器则集中了所有的应用逻辑。

5、操作简单

C/S 多是建立在 Window 平台上,表现方法有限,对程序员普遍要求较高。B/S 建立在浏览器上,有更加丰富和生动的表现方式与用户交流,并且大部分难度减低,降低开发成本。

本 RFID 中间件采用 B/S 体系架构, B/S 结构分布性强,维护和升级简单,采用 B/S 构架开发中间件,与 RFID 的分布式应用相吻合。但同时 B/S 构架也存在一些问题。如数据安全性问题、数据传输速度慢、软件的个性化特点降低等。虽然可以用 ActiveX、Java 等技术开发较为复杂的应用,但是相对于发展已非常成熟 C/S 的一系列应用工具来说,这些技术的开发复杂,并没有完全成熟的技术工具供使用。

3.3 SOAP 传输协议

SOAP (Simple Object Access Protocol) 简单对象访问协议是在分散或分布式的环境中交换信息的简单的协议,是一个基于 XML 的协议, SOAP 是一种轻量级协议,用于在分散型、分布式环境中交换结构化信息。SOAP 利用 XML 技术定义一种可扩展的消息处理框架,它提供了一种可通过多种底层协议进行交

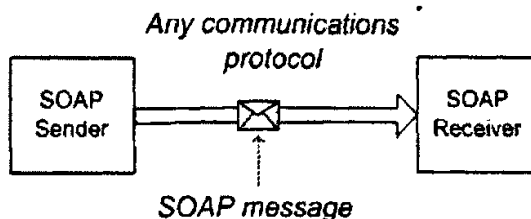
换的消息结构 j 。这种框架的设计思想是要独立于任何一种特定的编程模型和其他特定实现的语义。它包括四个部分：SOAP 封装(envelop), 封装定义了消息中的内容是什么, 是谁发送的, 谁应当接受并处理它以及如何处理它们; SOAP 编码规则 (encoding rules), 用于表示应用程序需要使用的数据类型的实例; SOAP RPC 表示(RPC representation), 表示远程过程调用和应答的协定; SOAP 绑定 (binding), 使用底层协议交换信息。

SOAP 消息基本上是从发送端到接收端的单向传输, 但它们常常结合起来执行类似于请求 / 应答的模式。所有的 SOAP 消息都使用 XML 编码。一条 SOAP 消息包含有一个必需的 SOAP 的封装包, 一个可选的 SOAP 标头和一个必需的 SOAP 体块的 XML 文档。

SOAP 提供的消息处理框架有以下特点:

可扩展性 简单性总是比效率和纯技术更重要, 因而互操作性成败的关键, 就在于必须绝对要求简单性。简单性仍然是 SOAP 的主要设计目标之一, SOAP 定义了一种通信框架, 允许以分层扩展的形式随着时间推移而加入这些特性。Microsoft、IBM 和其他软件厂商正在积极开发一个 SOAP 扩展的通用套件, 该套件将加入大多数开发人员期待的特性。这一计划被称为全局 XML Web 服务结构 (GXA)。Microsoft 已经发布了针对若干 GXA 规范的一个参考实现, 并将其命名为 Web Services Enhancements 1.0 SP1 for Microsoft .NET (WSE)。

可在任何传输协议 (TCP、HTTP、SMTP, 甚至是 MSMQ) 上使用 (参见图 1)。但是为了保持互操作性, 需要定义一些标准协议绑定, 以便草拟用于每种环境的规则。SOAP 规范提供了一种用于定义任意协议绑定的灵活框架, 并且由于 HTTP 的使用极为广泛, 它现已为 HTTP 提供了一种显式绑定。



允许任何编程模型, 并且不依赖于 RPC。基本的 SOAP 模型更接近于传统的消息处理系统, 如 MSMQ。SOAP 定义了一种模型以便处理个别的单向消

息。可以将多条消息组合成一条整体的消息交换。SOAP 允许使用任何数量的消息交换模式 (MEP), 包括请求/响应、要求/响应 (与请求/响应相对)、通知和长期运行的点对点对话等。鉴于 RPC 的广泛应用, SOAP 草拟了一项协议, 以便将 RPC 用于 SOAP。

随着计算机技术的不断发展, 现代企业面临的环境越来越复杂, 其信息系统大多数为多平台、多系统的复杂系统。这就要求企业解决方案具有广泛的兼容能力, 可以支持不同的系统平台、数据格式和多种连接方式, 要求在 Internet 环境下, 实现系统是松散耦合的、跨平台的, 与语言无关的, 与特定接口无关的, 而且要提供对 Web 应用程序的可靠访问。

本文提出的 RFID 中间件解决方案把 SOAP、HTTP 和 RPC 结合起来, 采用 HTTP 作为底层通讯协议; RPC 作为一致性的调用途径, XML 作为数据传送的格式, SOAP 语义自然映射到 HTTP 语义, RPC 请求映射到 HTTP 请求上, 而 RPC 应答映射到 HTTP 应答。同时利用了 SOAP 的可扩展, 可通用的特点以及 HTTP 的丰富的特征库的优点, 把成熟的基于 HTTP 的 WEB 技术与 XML 的灵活性和可扩展性组合在一起, 使得系统能够无缝地进行通信和共享数据, 在 Internet 环境下, 消除巨大的信息孤岛, 实现信息共享、进行数据交换, 达到信息的一致性。实现不同的系统之间能够用“软件-软件对话”的方式相互调用, 打破了软件应用、网站和各种设备之间的格格不入的状态, 实现“基于 WEB 无缝集成”的目标。

第四章 RFID 中间件各功能模块的设计与实现

通过前面几章的分析, 对 RFID 中间件的功能和整体结构有了清楚的了解, 系统分为四部分: 控制中心、硬件管理器、事件处理器、硬件管理器。这四部分相互协作完成中间件的功能。本章主要讨论这四部分的设计与实现。

4.1 控制中心

4.1.1 功能描述

控制中心分为两个功能模块：系统管理、配置管理。

系统管理功能模块具体包括以下子功能：

1) 操作员登录。操作员用合法的帐号和密码登录控制中心，登录后的界面根据操作员的权限的不同而不同。

2) 添加操作员信息。给系统添加新的操作员（同时赋予相应的权限）。

3) 修改操作员信息。修改相应操作员的信息，包括基本信息、权限、密码等。

4) 查看操作员信息。根据给出的操作员登录名或相应的其它条件查看操作员信息。

5) 删除操作员信息。从系统中删除相应的操作员。

6) 退出系统。退出系统到登录界面。

配置管理模块包括以下功能：

配置管理。配置网关、license 信息。

配置事件管理器。配置事件处理器 `server` 的信息，包括逻辑读写器、监视器等信息。

配置硬件管理器。配置硬件管理器的信息，包括读写器、监视器等信息。

查看系统状态。查看各阅读器、事件处理器 `server` 及硬件管理器下读写器的状态。

控制中心通过网关与硬件管理器和事件管理器进行通讯，包括设置硬件管理器和事件管理器的基本参数，监视两者的状态以及与读写器进行交互。它们之间的交互如图 4-1：

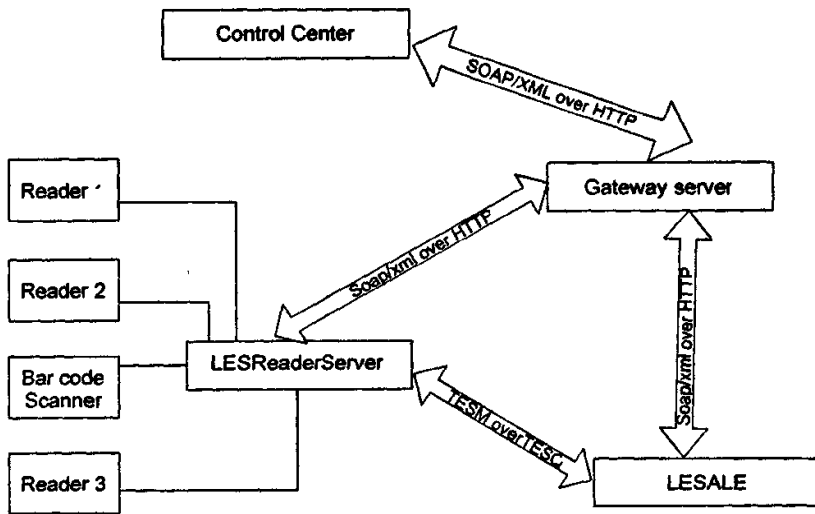


图 4-1 RFID 中间件功能模块交互图

4.1.2 接口描述

控制中心的主要功能是供操作员进行配置管理操作，基于 B/S 结构的设计，使得操作员只需用浏览器即可进行所有的管理工作，不需要安装任何的客户端软件。以下是几个主要的操作界面：

1、系统登录界面

该界面显示输入用户名、密码并登录上控制中心的界面。

欢迎光临xxx系统

用户名

密码

登录

重置

图 4-2 登录界面图

如果用户明或密码错误系统会提示出错，成功登陆界面如图 4-3：

2、登录成功界面

该界面由上面的头部显示系统信息、左边的操作菜单及右边的显示信息三部分组成，登录上控制中心后，显示的登录的页面的右边显示信息默认是查看

状态的信息。

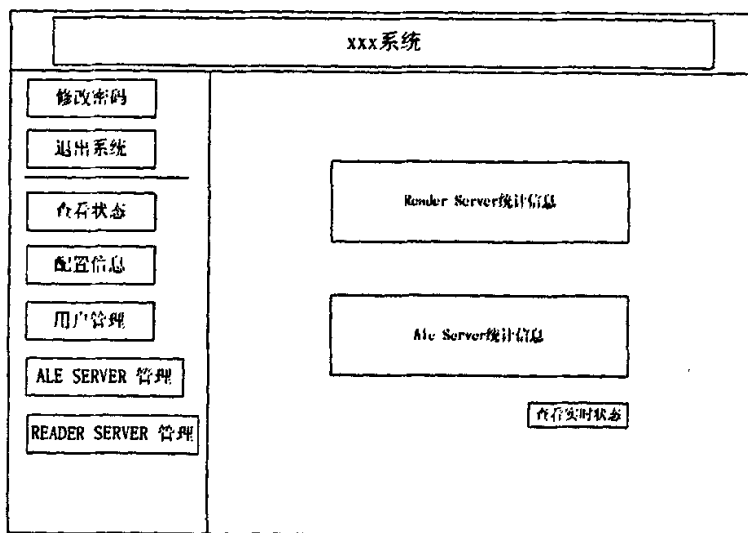


图 4-3: 登录成功页面

查看状态操作包括查看各服务器的状态以及服务器下各个识读器的状态；配置信息操作包括基本配置及授权配置。默认是基本配置，包括配置网关、端口等。

3、用户管理界面

用户管理包括界面如图 4-4:

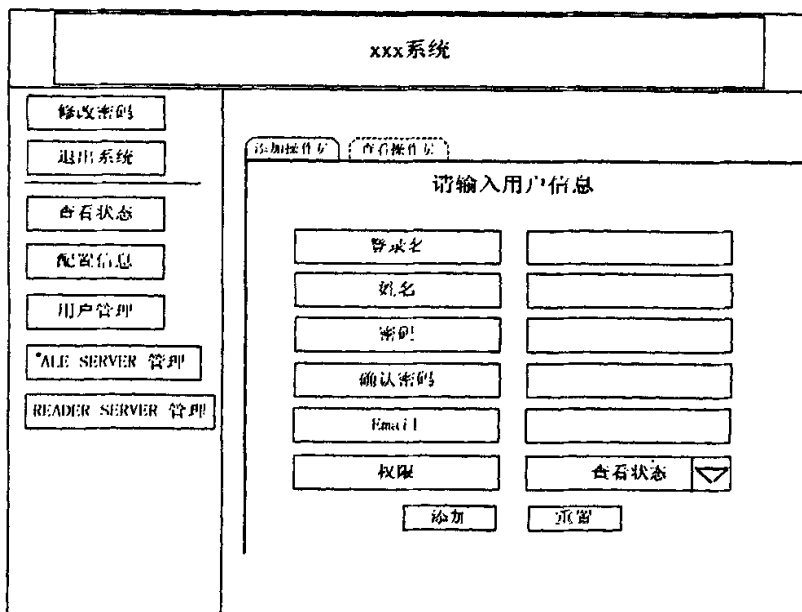


图 4-4 用户管理界面

该界面有三个操作：添加、查看用户，查看用户按列表列出系统中所有的用户。在查找到相应的用户及列表显示用户后都可做修改、删除用户信息操作，用户管理的界面默认是到添加用户的操作。权限的下拉列表有两项：查看状态、所有操作。具有查看状态权限的操作员登录后只有查看系统的状态图，具有所用操作权限的操作员具有所有操作的权限。

4、硬件管理器管理界面 如图 4-5：

图 4-5 硬件管理器管理界面

硬件管理器管理界面用来管理硬件管理器本身以及它连接的读写器，默认是管理器的配置界面，打开读写器按钮后，出现硬件配置界面，如图 4-6：

5、读写器配置界面

名称	当前数据来源	当前数据信道	操作
READER_1	XXXX	XXXX	删除
READER_2	XXXX	XXXX	删除
READER_3	XXXX	XXXX	删除

图 4-6 读写器信息的界面

主要的接口函数如表 4-1:

函数名称	功能描述	参数	返回值
UpdateReaderServerConfigure()	重新配置硬件管理器	xml 格式的 配置文件 字符串	int
UpdateALEServerConfigure()	重新配置事件管理器	xml 格式的 配置文件 字符串	int
StartAllReader ()	启动所有的阅读器	null	int
StopAllReader ()	停止所有的阅读器	null	int
GetReaderServerStatus ()	获得硬件管理器的状 态信息	null	int
GetALEStatus()	获得事件管理器的状 态信息	null	int
GetReaderServerConfigure()	获得硬件管理器的配 置信息	null	硬 件 管 理 器 配 置 文 件 的 字 符 串
GetALEServerConfigure()	获得事件管理器的配 置信息	null	事 件 管 理 器 配 置 文 件 的 字 符 串

表 4-1 控制中心的接口函数

4.2 硬件管理器

根据 EPC global Working Draft 的 Reader Management 1.0, 读写器应该有很强大的功能。而现在实际厂商并没有按这个标准做出相应的硬件管理器, 并且现在的读写器的可靠性也较差, 种类繁多。因此需要一套可用的 RFID 的中间件, 使用户不必关心物理的读写器, 只需通过这套中间件系统控制读写器, 过滤和处理接收读写器发送的消息, 发送给应用程序。

4.2.1 功能分析

与操作员直接打交道的是控制中心, 操作员管理硬件设备的操作通过操作控制中心实现 (通过浏览器), 控制中心再通过硬件管理器管理硬件, 因此硬件管理器要具有响应控制中心的命令功能。事件处理器会向硬件管理器发送命令,

如多长时间采集一次数据等, 因此硬件管理器还要有与事件处理器交互的功能。

综上所述, 硬件管理器的功能分为两大部分:

1、响应控制中心和事件处理器的命令。如表 4-2:

功能	子功能
响应控制中心发送的命令	启动所有读写器
	停止所有读写器
	更新配置文件
	获取硬件管理器状态
响应事件处理器发送的命令	增加过滤器
	删除过滤器

表 4-2 硬件管理器的相应功能

2、操作硬件设备, 收集原始数据, 并对数据进行初步处理。如表 4-3:

功能	描述
数据采集	通过读写器或条码扫描等底层设备采集符合 EPC global 规定的标签数据。
数据过滤	对采集的数据进行过滤。
读写器状态监控	收集读写器状态信息生成相应报告传送到硬件管理器, 由硬件管理器传输到控制中心。
事件处理	通过对标签状态的监测, 生成不同的事件报告并进行过滤,

表 4-3 硬件管理器的处理功能

根据硬件管理器的功能, 把它分为两个模块: 接口模块和处理模块, 其中接口模块负责接收其它模块发送的命令和信息, 并根据命令的类型执行相关的操作 (包括调用处理模块的操作); 处理模块负责与具体的硬件设备交互并向接口模块提供接口。

4.2.2 接口模块

接口模块的工作流程图如图 4-7, 在接口模块启动后, 会一直监听消息。控制中心在启动时会发送命令获取配置文件, 如果配置文件不存在, 则接口模块

返回错误，控制中心也不会让用户继续启动硬件管理器。用户停止硬件管理器时，配置对象并不会被销毁，这样下次用户启动时就不用重新读取 XML 文件生成配置对象。

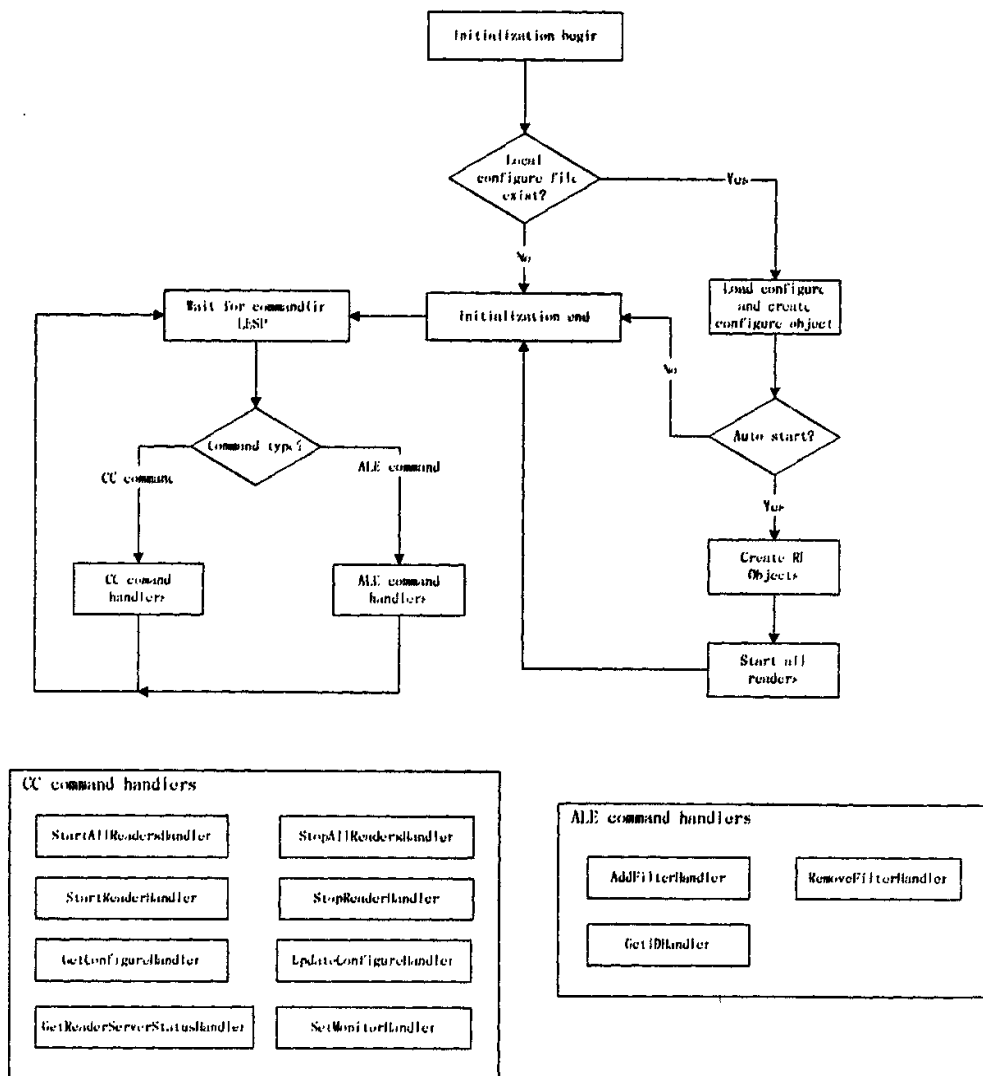


图 4-7 接口模块工作流程图

当用户更新硬件管理器配置时，在控制中心端按 Update 按钮，控制中心会把全部的硬件管理器配置信息保存为 XML 文件，发送给网关，网关转发给硬件管理器，硬件管理器接口模块获得 XML 文件，将文件写入磁盘，覆盖原来的 XML 配置文件。销毁接口模块进程中的配置对象，然后解析配置文件，生成对应的配置对象，等待用户启动所有阅读器；当控制中心要结束所有阅读器时，发送停止命令给网关，网关转发给硬件管理器，硬件管理器接口模块获得

命令字，结束 Read Server 进程；控制中心发送启动命令给网关，网关转发给硬件管理器，硬件管理器 接口模块获得命令字，根据配置对象调用 RP 提供的接口创建处理模块对象（这个过程参见创建处理模块对象流程图），启动所有阅读器。其它命令基本遵循同一模式，接口模块接收到消息，执行消息对应的动作，这个动作一般只需要一个简单的函数调用即可完成。接口函数如表 4-4：

函数	功能
StartAllReaders()	启动所有 reader
StopAllReaders()	停止所有 reader
StartReader(conststd::string &name)	启动名称为 name 的 reader
StopReader(conststd::string &name)	停止名称为 name 的 reader
GetRSConfig()	启动返回所有 reader 的配置信息
UpdateRSConfig(conststd::string &xmlString)	把控制中心发送过来的 xml 文件存储在本地
GetRSStatus()	返回 ReaderServer 当前的状态

表 4-4 硬件管理器接口函数

4.2.3 处理模块

Reader Protocol 是 EPC Global 为 Reader 生产厂商指定的一套关于数据采集过滤的协议规范，通过此规范的定义使大家都在 tag 数据的收集、过滤、传输能够采取相同的接口标准，方便事件处理器系统的扩展和物联网的建立。

Reader Management 是针对符合 EPC Global 标准的 Reader 设备进行监控管理而制定的协议规范，根据这个规范我们制定相关的功能接口以便控制中心对读写器设备进行管理。

为了使本中间件与 EPC Global 的标准兼容，我们根据 RP 和 RM 协议规范的草稿，设计并实现我们的中间件。

处理模块分为三个层次，依次为：数据收集层、消息层、数据传输层。最底层是数据收集层，主要负责底层数据的采集，再往上是消息层，负责消息的过滤、格式转换等处理，最高层是数据传输层，负责把处理过的数据传输给上级。图 4-8 为处理模块的层次结构：

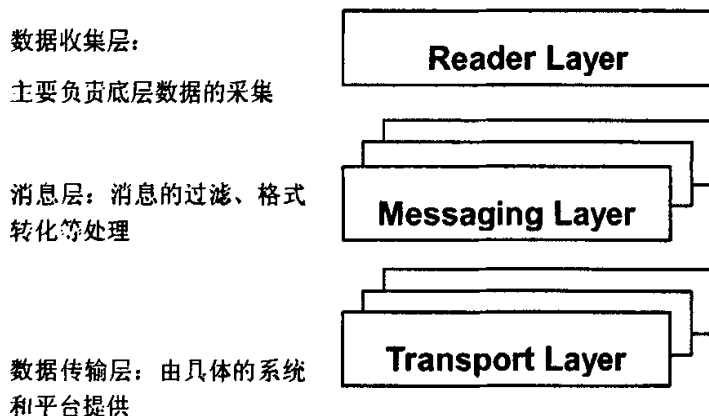


图 4-8 硬件管理器处理模块层次结构图

1、数据收集层

数据收集层的设计目标是为整个系统提供精确的实时数据。整个系统的可用性、可靠性等都以此为基础。主要包含阅读器的管理、大规模阅读器间的协调、异构阅读器网络管理等功能。

2、消息层

消息层是处理模块的核心。RFID 事件处理以数据挖掘、神经网络、传感网络、复杂事件处理等理论为基础^[23]，针对原始数据规模大和原始数据包含的语义信息少两个问题，来有效减少数据冗余、压缩事件规模并为上层商业应用提供语义信息。图 4-9 给出了事件处理的过程：

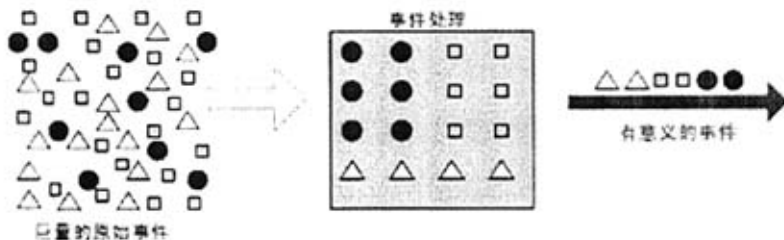


图 4-9 RFID 事件处理过程

RFID 事件处理主要研究内容包括事件描述、事件过滤、事件挖掘、事件聚合、事件存储等。

1、事件处理模型 事件是“指示某种行为的信息”[5]，包括系统产生的消息，系统状态的改变，任务的开始和结束等。事件在形式上类似于消息，如都

包含数据, 不同在于事件直接指示某些行为的发生。事件根据角度不同有多种分类方法。根据事件语义的聚合程度不同, 事件可以分为简单事件和复杂事件。从分层的角度划分, 事件又可分为底层事件和高层事件, 底层事件是系统产生的实际事件(actual event); 高层事件是由用户自定义的, 从底层事件映射而成的复杂事件(虚拟事件), 复杂事件包含了更多的语义信息。从系统响应的角度来划分, 事件可以分为常规事件、异常事件等。事件处理是指从原始事件生成复杂事件的过程, 其中包含事件描述、过滤、挖掘、聚合、响应、存储等操作^[24], 已有的事件处理模型主要包括 ECA(Event-Condition-Action), CEP(Complex EventProcessing), Situation Manager 等。

2、事件过滤 它是在输入的巨量事件中发现有用的和重要的事件, 过滤冗余的、无关的数据, 其目标在于减小事件的数量。在 RFID 事件过滤方面, 尚无成熟的过滤规则或标准可以遵循。Sun 和 SAP 定义了一些简单过滤规则, 如 smooth、delta 等。尽管商业逻辑不同, 但过滤都可归结为一些特定操作, 如分组、计数、冗余删除、区分等。

3、事件挖掘 事件挖掘是基于事件之间的时间、空间和因果关系以及事件的属性信息, 利用形式化的模式语言, 实时地从大规模事件集合中提取模式的过程。这个过程所发现的模式是事件聚合的基础。事件挖掘是数据挖掘在复杂事件处理研究领域的延伸。

4、事件聚合 它是指由匹配某种模式的事件子集, 生成符合相应输出模式的高层事件的过程。该事件通常具有更丰富的语义信息, 更易于被应用程序所理解和使用。

5、事件存储 其当前研究的主要目标是为了更高效地处理大批量事件数据, 减少数据处理中对后台数据库的频繁操作和因存储、查询所带来的数据在网络中的来回传输。其中, 内存数据库的研究是当前热点, 内存数据库采用不同的缓存策略, 使 RFID 系统在把数据提交到磁盘存储之前, 把数据写到 RAM 中去, 其操作效率是传统操作效率的几百倍甚至几千倍。

3、数据传输层

从消息层传递来的 RFID 信息流, 不同的应用有不同的计算需求, 如物流领域用来定位与追踪, 安全领域用于身份识别, 终端客户用于物品防伪等。但是

不同的应用都有信息存储、信息包的路由、信息发布、访问控制、安全认证等共性需求，这些共性需求可抽取出来作为支撑不同应用的基础设施。由这些基础设施就构成了整个信息应用层。

4.3 事件处理器

4.3.1 ALE

从 RFID 标签制造开始，到其信息被 RFID 阅读器捕获，再由 RFID 中间件进行事件过滤和汇总，然后由 EPCIS 应用软件进行 RFID 事件的业务内容丰富，保存到 EPCIS 存储系统，供企业自身和其合作伙伴进行访问，这其中的各个环节，EPCglobal 都出台了相关标准和规范。关于 RFID 中间件，EPCglobal 于 2005 年 9 月推出了事件处理器（应用级别事件，Application Level Event）规范，针对 RFID 中间件和 EPCIS 捕获应用之间，定义了 RFID 事件过滤和采集接口（事件处理器）。

事件处理器层介于应用业务逻辑和原始标签读取层之间。它接收从数据源（一个或多个读写器）中发来的原始标签读取信息，而后，按照时间间隔等条件累计（Accumulate）数据，将重复或不感兴趣的 EPCs 剔除过滤(Filter)，同时可以进行计数及组合（Count/Group）等操作，最后，将这些信息对应用系统进行汇报。

例如：Reader1 在 t_1 时刻读到 tag1,可简记为 (Reader1, t_1 ,tag1) 三元组。由于在 t_1 时刻，Reader1 可以读到很多标签，所以它将向事件处理器层输入一组三元组 (Reader1, t_1 ,tag1), (Reader1, t_1 ,tag2)}，在 t_2 时刻，如果 tag2 离开 Reader1，tag3 进入 Reader1，那么读写器将向 LE 层输入{ (Reader1, t_2 ,tag1), (Reader1, t_2 ,tag3)}。这些读写器输出的分散的标签读取信息数据量很大，需要进行过滤和收集，告诉上层应用系统什么事件发生，事件处理器层的主意任务就在于此。经过事件处理器处理之后，底层简单的读取标签信息流将变成 t_2 时刻 tag2 离开 Reader1, t_2 时刻 tag3 进入 Reader1,简记为(Reader1, t_2 , tag2, DELETE), (Reader1, t_2 , tag3, ADD)。通过这些事件的输出，上层业务逻辑将获得良好的应用接口。

在事件处理器中，应用系统可以定义这些内容：在什么地方（地点可以映射一个或多个读写器及天线）读取标签。在怎样的时间间隔内（决定时间、某个外部事件触发）收集到的数据，如何过滤数据，如何整理数据报告内容（按照公司、商品还是标签分类），标签出现或消失时是否对外报告，以及读取到的标签数目。

在事件处理器模型中，有几个最基本的概念：读周期（Read Cycle），事件周期（Event Cycle）和报告（Report），如图 4-10 所示。“读周期”是和读写器交互的最小单位。一个读周期的结果是一组 EPCs 集合。读周期的时间长短和具体的天线、RF 协议有关。读周期的输出就是事件处理器层的数据来源。“事件周期”可以是一个或多个读周期。它是从用户的角度来看待读写器的，可以将一个或多个读写器当作一个整体，是事件处理器接口和用户交互的最小单位。应用业务逻辑层的客户在事件处理器中定义好事件周期的边界之后，就可接收相应的数据报告。“报告”则是在前面定义的事件周期的基础上，事件处理器向应用层析提供的数据结果。

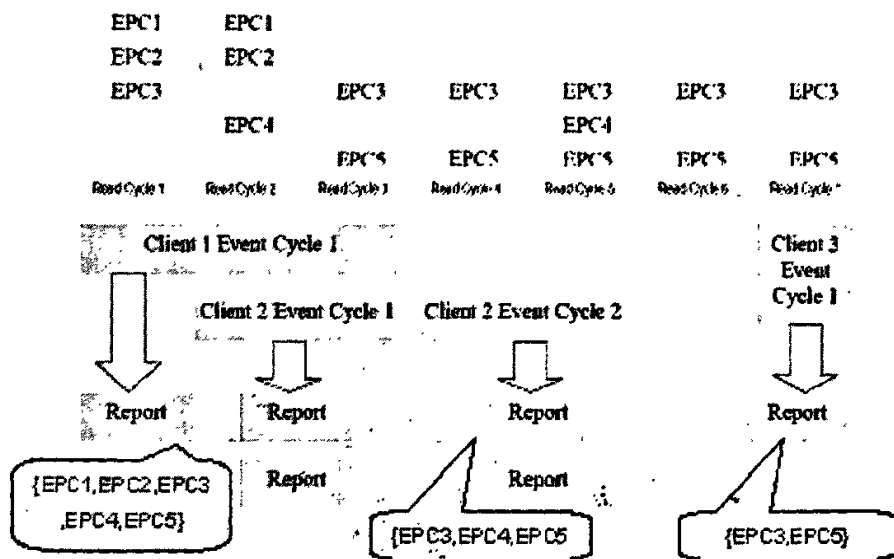


图 4-10 ALE 模型

每个读周期，RFID 中间件会发送读取 RFID 标签指令给读写器，读写器控制读写天线发送电磁信号，一旦 RFID 标签内置天线在其信号工作范围内获得这一电磁信号，会激活其芯片上的信息发送逻辑，发送电磁响应，读写器的读

写天线就会捕获到 RFID 标签信息, 发送给 RFID 中间件。这个读周期是可控的, 可由 RFID 中间件进行配置管理。

一个事件周期会包含一个或多个读周期, 因此, RFID 中间件在这一事件周期内读取的所有标签信息被聚合起来, 然后通过过滤机制, 发布成为一个 ECRReport。ECSpec 用来描述事件周期间隔或事件触发机制, ECRReport 的格式, 事件过滤和组合规则等。ECRReport 是 EPC 事件的汇集, 还包括其它相关的信息, 譬如逻辑阅读器名称, ECSpec 信息等。

ECSpec 则描述了事件周期以及报告产生的格式。它包括: 一组逻辑读写器 (logical Readers) 内, 这些逻辑读写器的读周期在该事件周期内; 一份定义事件周期边界的规范; 以及在这个事件周期内产生的一组报告 (report) 的格式规范。

4.3.2 事件处理器的实现

事件处理器由控制中心启动, 控制中心向事件处理器发送配置文件, 根据配置文件创建 logic reader, 同时监听控制中心发送的信息, 当控制中心发送消息时, 事件处理器 manager 会收到这个消息, 并根据命令调用相应的函数。事件处理器启动流程如图 4-11:

上层应用系统通过调用事件处理器接口获得处理后的 EPC 数据, 应用系统与事件处理器中间件的交互, 必须先将事件周期的定义文件 (ECSpec) 传送到中间件, 同时告知中间件将报告发回的地址。在以事件处理器交互中, 有几个最基本的方法: define/undefine, subscribe/unsubscribe, poll/Immediate。其中 define/undefine 是定义/撤销 ECSpec 的操作, subscribe/unsubscribe 是订阅/撤销某个 ECSpec 的服务。有三种交互方式: 直接交互, 间接交互, 以及 Poll/Immediate

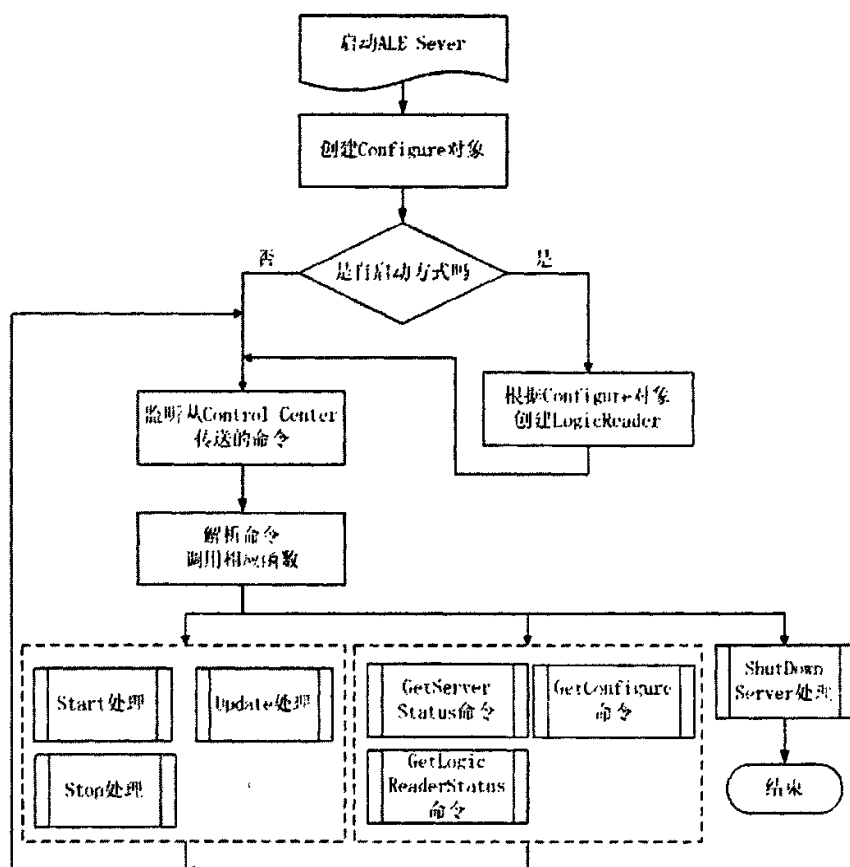


图 4-11 事件处理器的启动流程

1、直接交互

该模式下，ECSpec 由客户 A 定义，得到的报告反馈给 A。首先，Client1 将名为 ECName1 的 ECSpec 定义给事件处理器中间件，而后 Client1 订阅该 ECName1 的报告，并将它发至地址为 notifyURI 的接收处。

在时间 1 内，读写器 reader1 没有读到标签，所以没有反馈。在时间 2 内，读到标签，而后，事件处理器中间件自动将 ECreport 发送给 Client1。

当 Client1 不需要 RFID 信息时，它首先退订 notifyURI 的 ECName1 的服务。当 ECName1 没有订阅者之后，就可以撤销 ECName1 的时间周期。流程图如 4-12：

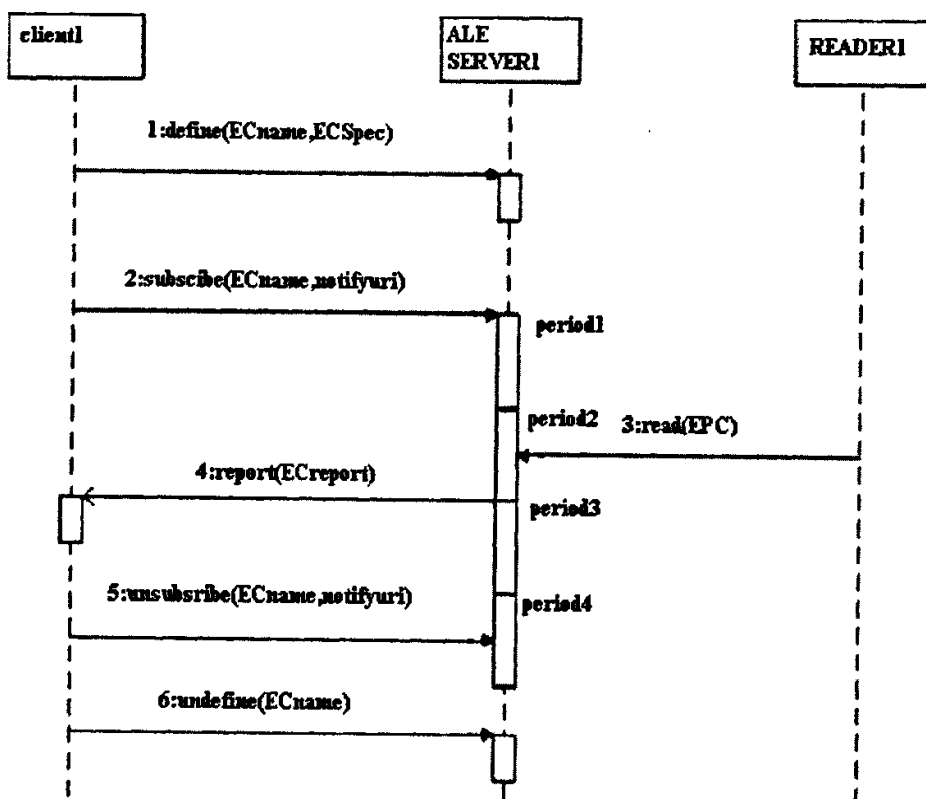


图 4-12 直接交互流程图

2、间接订阅

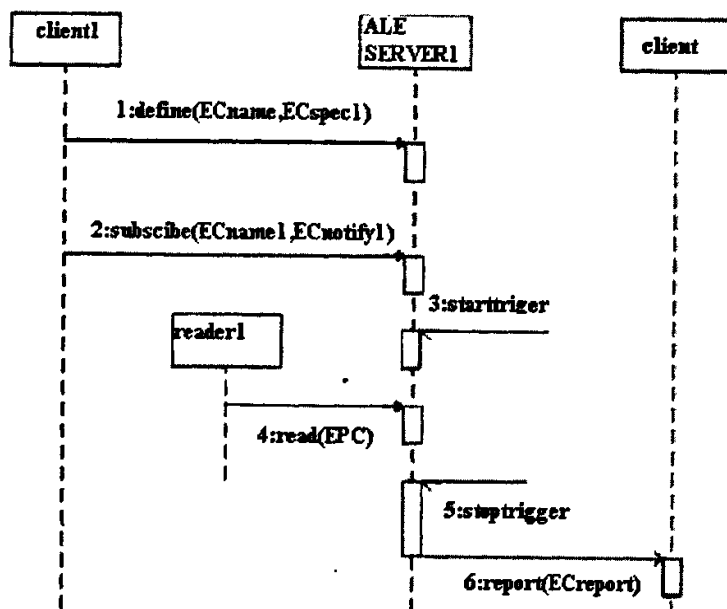


图 4-13 间接交互流程图

该模式与直接订阅的差异是，得到的报告不是反馈给 A，而是反馈给 B。

该图显示的 ECspec 边界由触发器来决定。在第 6 步中，我们可以看到 ECreport 发至 client1，而不是初始的服务定义者。这是由于在第 2 步中的服务反馈地址 notifyURI 指向 client1。

3、Poll/Immediate

Poll 和 Immediate 可以看成应用系统对事件处理器中间件的快照。在很多应用中，不需要一直监听事件处理器，而只要知道当时读到的标签信息，这两种模式就是为满足这些需求而设计的，流程图如 4-14：

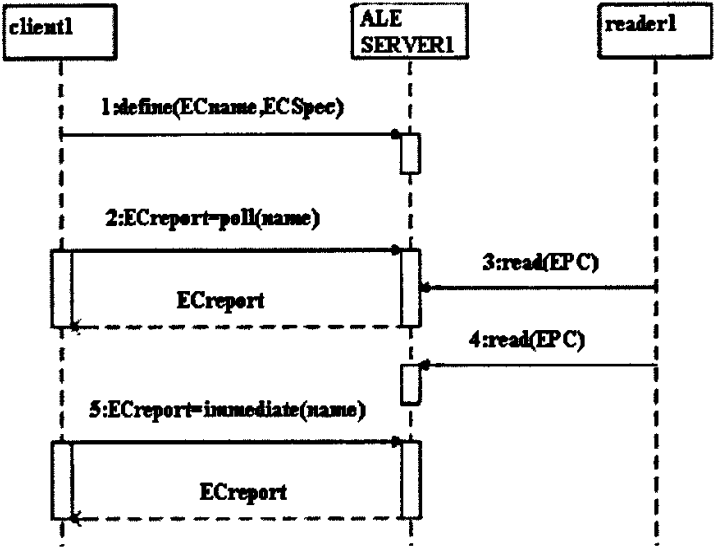


图 4-14 Poll 和 Immediate 流程图

当事件处理器中间件中已经有定义好的 ECSpec 时，同时 Client 需要这个 ECSpec 提供的信息，就可以使用 Poll 方法得到反馈。

当事件处理器中间件中不存在 Client 需要的事件周期时，这个时候，可以直接转送这个事件周期的定义 ECSpec2，而后得到结果，这就是 Immediate。

事件处理器与上层应用系统之间的交互函数如表 4-5：

函数名	功能
define(specName: String, spec: ECSpec)	定义一个 ECSpec，并指定一个名字
undefine(specName: String)	通过 ECSpec 名字，取消 ECSpec
getECSpec(specName: String)	根据一个名字返回一个 ECSpec 对象
getECSpecNames()	返回一个 specNames 的字符串列表
subscribe(specName: String, notificationURI: String)	订阅一个 ECSpec，相应的 ECreports 被发送到指定的 URL。

<code>unsubscribe(specName:String, notificationURI: String)</code>	取消一个 ECSpec 订阅, 停止向指定的 URL 发送 ECRports。
<code>poll(specName: String)</code>	指定一个 specName, 同步返回一个 ECRport
<code>immediate(spec: ECSpec)</code>	指定一个 ECSpec, 同步返回一个 ECRport。
<code>getSubscribers(specName: String)</code>	Return all notification URLs for the specified ECSpec Name
<code>getStandardVersion()</code>	Return the standard version that is implemented
<code>getVendorVersion()</code>	Return TES 事件处理器 version info

表 4-5 事件管理器与上层应用的接口

4.4 网关 (Gateway)

4.4.1 xml-rpc 的实现方式

网关是连接中间件个部分的纽带, 各部分之间信息的传输都通过网关实现,

Gateway 采用 xml-rpc 来实现, xml-rpc 允许运行在不同操作系统、不同环境的程序实现基于 Internet 过程调用的规范和一系列实现。这种远程过程调用使用 http 作为传输协议, xml 作为传送信息的编码格式。

xml-rpc 分客户端与服务器端, 在 Gateway 实现的是服务端, 服务器端可运行在 apache xml-rpc 的内建的 http server 或运行在 servlet 环境中。

本中间件的 xml-rpc 是运行在 tomcat 的 servlet 中, 典型的代码如下所示:

1. 建立 XmlRpcFacade

```
package helloxmlrpc;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.OutputStream;
import org.apache.xmlrpc.XmlRpcServer;

public class XmlRpcFacade {
    private static XmlRpcServer xmlrpc;
    static{
        xmlrpc = new XmlRpcServer(); //登记 handler object.
```

```

        xmlrpc.addHandler("hello", new HelloHandler());
    }

    public void execute(HttpServletRequest request, HttpServletResponse
response) throws
        IOException {
        byte[] result = xmlrpc.execute(request.getInputStream());
        response.setContentType("text/xml; charset=GB2312");
        response.setContentLength(result.length);
        OutputStream out = response.getOutputStream();
        out.write(result);
        out.flush();
        out.close();
    }
}

```

2、建立 servlet

```

package hellpxmlrpc;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;

public class XmlRpcServlet extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=GBK";
    private XmlRpcFacade facade;

    public void init() throws ServletException
    {
        facade = new XmlRpcFacade(); //处理 HTTP Get request
    }

    public void doGet(HttpServletRequest request, HttpServletResponse
response) throws
        ServletException, IOException {
        this.doService(request, response);
    }

    public void doPost(HttpServletRequest request, HttpServletResponse
response) throws

```

```

        ServletException, IOException {
            this.doService(request, response);
        }
        public void doService(HttpServletRequest request,
                                HttpServletResponse response) throws
ServletException,
                                IOException {
            facade.execute(request, response);
        }
        public void destroy() {
        }
    }
}

```

4.4.2 Gateway 的数据结构设计

在分布式应用中，一般会有多个硬件处理器和多个事件处理器共存，Gateway 需要知道控制中心发下的命令是发给哪个硬件处理器或事件处理器的，在 Gateway 建立一张映射表，映射关系是 Resource ID 与 Resource URI 的关系。比如系统中有两个 reader server 他们的 Resource ID 及 Resource URI 分别是表 4-6 所示：

Resource ID	Resource URI
1111	http://10.0.0.1:8080
1112	http://10.0.0.2:8080
1113	http://10.0.0.3:8080
1114	http://10.0.0.4:8080
1115	http://10.0.0.5:8080

表 4-6 硬件处理器映射表

实现时在数据库的对应表中存放相应记录，而 Resource ID 与 Resource URI 的对应关系由控制中心对硬件管理器及事件处理器配置时把对应系统传到 Gateway 中来。

第五章 结论和展望

本文研究设计了 RFID 系统的总体方案, 设计了基于 ALE 的 RFID 中间件系统模型和框架结构, 本设计是基于开放式架构设计的、模块化的、可升级的数据处理系统。由计算机硬件以及系统数据处理软件构成。主要用来加工和处理来自读写器的所有信息和事件流的软件, 是连接读写器和企业应用的纽带。

本设计有以下几个创新点:

(1) 独立于架构 RFID 中间件独立并介于 RFID 读写器与后端应用程序之间, 并且能够与多个 RFID 读写器以及后端应用程序连接, 以减轻架构与维护的复杂性。

(2) 数据流 RFID 的主要目的在于将实体对象转换为信息环境下的虚拟对象, 因此数据处理是 RFID 最重要的功能。RFID 中间件具有数据的搜集、过滤、整合与传递等特性, 以便将正确的对象信息传到企业后端的应用系统。

(3) 处理流 RFID 中间件采用程序逻辑及存储再转送的功能来提供顺序的消息流, 具有数据流设计与管理的能力。

本设计目的是要推出可伸缩的 RFID 中间件框架, 在此基础上, 用户可以更有效地进行开发和适应变化的业务需要。本 RFID 中间件能操纵控制 RFID 读写设备按预定方式工作, 保证不同读写设备之间很好地配合协调, 按照一定的规则过滤数据, 筛除绝大部分冗余数据, 将真正有效的数据传送给后台的信息系统, 并且兼容各种符合 EPC 标准的 RFID 硬件设备; 用户可以围绕 RFID 进行业务流程创新, 开发新应用; 与现有应用系统集成, 产生真正的商业价值。

本设计需要完善和改进的几个关键问题:

查询服务的安全性、网络信息安全性是本设计需要完善和改进的问题。随着 RFID 技术的推广和使用, 由于 RFID 的分布特性, 更加上交易量的不断增加, 系统在伸缩性、可用性、安全性和互操作性将面临着挑战。

总之, 本设计根据 EPC 物联网的结构以及物流和产品信息管理在现代的发展, 选择合理、优化的技术手段, 适合企业在生产、销售、流通的等领域的现代化管理, 并能降低企业管理成本、提高效率、创造巨大的经济效益。

参考文献

- [1] 微软公司. Microsoft Windows CE Device Driver Kit 设备驱动程序开发指南[M]. 北京: 希望电子出版社, 1999.
- [2] [德] Klaus Finkenzeller. 射频识别(RFID) 技术. 北京: 电子工业出版社, 2002.
- [3] Clark S, Traub K, Anarkat D, et al. Auto-ID Savant Specification 1.0[R]. Auto-ID Center, 2003.
- [4] 钟慈安. RFID 技术运作的神经中枢——RFID Middleware[J]. Electronic Commerce Pilot, 2004, 6(14).
- [5] Luckham D C, Frasca B. Complex Event Processing in Distributed Systems[R]. Stanford University, Technical Report: CSL-TR-98-754, 1998.
- [6] 丁振华, 李锦涛, 冯波, 郭俊波. RFID 中间件研究进展
- [7] 刘丽华, 董天临. 基于以太网接口的 RFID 中间件软件设计, 单片机与嵌入式系统应用, 2006 年第 7 期
- [8] 游战清, 李苏剑等. 无线射频识别技术(RFID)理论与应用. 北京电子工业出版社
- [9] Christian Floerkemeier, Matthias Lampe. RFID middleware design addressing application requirements and RFID constraints.
- [10] 陈十一. EPC 全球展望和中国的应对策略. 第二界 EPC 与物联网高层论坛会, 上海. 2004: 7~12
- [11] EPC 应用分析及各国发展情况与中国的应对策略. EPC 与物联网高层论坛办公室, 北京. 2004: 8~10
- [12] 唐莉. 一个支持 JMS 的消息中间件系统的设计与实现, 吉林大学硕士论文, 2003
- [13] 张云勇, 张智江, 刘锦德等. 中间件技术原理与应用, 清华大学出版社, 2004. 10
- [14] Landt J, Catlin B. Shrouds of Time: The History of RFID[R]. AIM Inc., 2001.
- [15] Savant 概要介绍. 中国物品编码中心. 2004: 10~13
- [16] 产品电子编码方案—电子产品码(EPC)研究课题报告 5. 中国物品编码中心. 2003: 8-9
- [17] Brock, D. L. The Electronic Product Code - A Naming Scheme for Physical Objects. Auto-ID White
- [18] Froning, G Pedrini, H J Tiziani et al. Vibration mode separation of transient phenomena using multi-pulse digital holography[J]. Opt Eng, 1999, 38(12): 2062~2068
- [19] Daniel W. Engels. A Comparison of the Electronic Product Code Identification Scheme & the Internet Protocol Address Identification Scheme. Technical Memo. 2002: 9~14
- [20] 李铁楠. 中间件在电子商务中的应用. 河北师范大学学报. 2002, 26(5): 8~12
- [21] <http://www.builder.com.cn/2002/0806/57323.shtml>
- [22] The Networked Physical World - Proposal for Engineering the Next Generation of Computing, Commerce and Automatic-Identification. Auto-ID White Paper, WH-001. 2000: 18~20
- [23] Etienne Roblet, Khalil Drira, Michel Diaz. Rormal design and development of a

- Corba-based application for cooperative HTML group editing support The Journal of Systems and Software 60 (2002) 113-127
- [24] Chue, P Marquet, C Depeursinge. Simultaneous amplitude- contrast and quantitative phase- contrast microscopy by numerical reconstruction of Fresnel off- axis holograms[J]. Applied Optics, 1999, 38(34) : 6994~7001
- [25] Brock, D. L . Intelligent Infrastructure - A Method for Networking Physical Objects. Presentation, MIT Smart World Conference. 2000: 8~10
- [26] Technology Guide. Auto-ID Center6, David Brock. The compact Electronic Product Code
- [27] 64-bit representation of the Electronic Product Code . Technical Report MIT-AUTOID-WH-008, MIT Auto-ID Center, Massachusetts Institute of Technology. 2001: 24~25
- [28] 张宏斌 例释 Java2 企业版 (J2EE) 程序设计, 中国铁道出版社, 2002. 8
- [29] 杨绍方, 深入掌握 J2EE 编程技术, 科学出版社. 2002. 5.
- [30] Anthony Lococo, David C, Yen Groupware. computer supported collaboration Telematics and Informatics 15(1998) 85-101
- [31] 胡啸, 陈星, 吴志刚, 无线射频识别安全初探. 信息安全与通信保密, 2005. 31
- [32] 朱士国, 基于 RFID 的计量管理系统研究. 合肥工业大学, 硕士论文, 2005: 43-45.

致 谢

本论文是在导师丁香乾教授的悉心指导和关怀下完成的，从最初选择课题、进行课题研究到最后撰写论文，在这些过程中无不都注入了导师丁香乾教授的心血。在近三年的求学生涯中，丁老师渊博的知识、严谨的治学态度和勇于开拓的工作作风使作者受益匪浅，导师的教诲和启发，学生将终生难忘。在此，谨向付出巨大心血和辛劳的导师表示至诚的敬意。

无论是在平时的学习生活中，还是学位论文的进行中，侯瑞春、杨宁老师、李建华博士等都给予了作者不少帮助和诚恳的建议，在此表示感谢。

作者要特别感谢家人，她们的支持和鼓励，给予了作者学习的动力和勇气；她们的悉心照顾，使作者能够顺利地完成学业！

最后，谨向所有给予作者关心、支持和帮助的老师、同学表示由衷的谢意。

攻读硕士学位期间完成的学术论文

- 1、郝兴贞，丁香乾. 无线射频识别（RFID）技术实现方式的研究与比较. 计算机应用研究。

攻读硕士学位期间获得的主要成果

1. 参与开发的“前置式 RFID 读写器一体机”，获得国家专利，专利申请号是：200610045332.7。