

RFID 轻量型中间件的构成与实现

林凤群¹, 陈伯成¹, 袁博¹, 刘文煌¹, 李晓燕¹, MAK Andy², 王洪洋²

(1. 清华大学深圳研究生院, 深圳 518055; 2. 深圳市讯流有限公司, 深圳 518055)

摘要: 为满足国内中小企业的实际管理需求, 设计一种射频识别(RFID)轻量型中间件, 其架构介于以基础架构为中心与以应用程序为中心的中间件之间, 简化了电子产品代码(EPC)系统的应用层事件结构。以应用程序接口为基础, 借鉴数据库集成过滤思想实现的轻量型中间件, 具有结构简单、开发周期短、易于后期维护和扩展的特点。

关键词: 射频识别; 轻量型中间件; 数据提取程序; 电子产品代码

Composition and Implementation of RFID Lightweight Middleware

LIN Feng-qun¹, CHEN Bo-cheng¹, YUAN Bo¹, LIU Wen-huang¹, LI Xiao-yan¹, MAK Andy², WANG Hong-yang²

(1. Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055; 2. Shenzhen Xunliu Co., Ltd., Shenzhen 518055)

【Abstract】 In order to satisfy the practice management requirement of inland small and medium enterprises, this paper designs a lightweight Radio Frequency Identification(RFID) middleware. Its architecture is between being centered with basic framework and centered with application program, and it simplifies the Application Level Event(ALE) structure of Electronic Product Code(EPC) system. Based on Application Program Interface(API), it implements lightweight middleware by integrating and filtering thought of database. This lightweight middleware has simple structure, short development cycle, and it is easy to maintain and extend at later period.

【Key words】 Radio Frequency Identification(RFID); lightweight middleware; data extraction program; Electronic Product Code(EPC)

1 概述

近年来食品安全问题频繁发生, 引起全世界广泛关注^[1]。很多发达国家和地区要求对出口到当地的食物能够进行跟踪和追溯, 无线射频识别技术(Radio Frequency Identification, RFID)为解决该问题提供了有效的技术途径, RFID中间件为多种读写器、多协议环境下工作的管理提供了高效支持, 本文是针对深圳输港蔬菜 RFID跟踪和追溯系统环境进行研制的。

RFID是一种非接触式的自动识别技术, 通过射频信号自动识别目标对象并获取相关数据, 无须人工干预, 工作于各种恶劣环境中, 可同时识别多个标签, 操作快捷方便。RFID系统主要由标签、阅读器和介于阅读器与企业应用之间的中间件3个部分组成。由于RFID目前尚未统一的标准, 造成许多RFID厂家生产的设备千差万别, 因此来自不同省份不同食品公司所运用的标签很可能存在差异。而RFID中间件的重要任务是屏蔽这些差异, 将RFID读写器读取的各种数据信息经过RFID中间件提取、解密、过滤、转换、导入企业的管理信息系统, 并通过应用系统反映在程序界面上, 供操作者监测与使用^[2]。

理论上, RFID在多种读写器、多协议环境下工作, 采用复杂的、功能齐全的、完全符合国际标准的、能管理读写器种类齐全的中间件是很重要的, 这意味着该软件可适应各种环境。目前, 流行RFID中间件的代表产品主要是基于EPCglobal标准, 这是比较完整的国际标准, 但其大而全、价格较昂贵、系统管理复杂, 且现国内服务支持尚不完善, 不太适合中小企业应用, 其不足主要从以下3个方面表现:

(1)重要性: EPC标准很重要, 但很多应用不需要复杂的

设备应用, 种类简单是系统集成成功的重要因素;

(2)发展趋势: 软件初建阶段旨在连接、沟通不同系统某个特定技术, 但通常最终其将成为多余;

(3)国内需求: 目前中小型企业需求是给出简化的思路或解决方案, 使企业以更少的成本、不依靠外部, 就能更可靠地解决和维护问题。

通常RFID中间件的应用较复杂, 不是接上任何读写器都能使用。基于此, 在国内企业实际管理条件下, 给出简化解决方案可能比研究出应用的实际中间件更重要。因此, 本文的轻量型中间件不是大型中间件的删减版本, 而是从满足国内中小型企业实际需求出发, 综合考虑实用性、简易型(好操作)进行的软件开发, 它拥有新的处理架构, 简单实用将是其主要特点。

本文研究的轻量型中间件实际上是一种简化的中间件, 构成方便, 又具有中间件的功能(跨各类读写器应用), 涉及几种常用的RFID读写器管理, 给出一个简单的系统框架, 输出输入信息符合EPCglobal标准, 使一些有编程基础的企业可以根据自己的特定需求构建轻量型中间件。

2 现有RFID中间件

RFID中间件在企业应用和读写器的数据采集之间扮演着

基金项目: 深圳市深港创新圈基金资助项目

作者简介: 林凤群(1985-), 女, 硕士研究生, 主研方向: 供应链与物流管理, RFID技术应用; 陈伯成, 副教授、硕士; 袁博, 讲师、博士; 刘文煌, 教授; 李晓燕, 副教授、硕士; MAK Andy, 工程师、硕士; 王洪洋, 博士

收稿日期: 2010-03-03 **E-mail:** chenbch@mail.tsinghua.edu.cn

很重要的角色。RFID 中间件是一种信息导向中间件(Message-Oriented Middleware, MOM)的软件中间件,信息是以消息的形式从一个程序模块传递到另一个或多个程序模块。由于消息可以非同步方式传送,传送者不必等待回应,因此 RFID 中间件包含的功能不仅是信息传送,通常还包含数据包分析与传播、安全保证、错误恢复、网络资源定位、路由选择、消息与要求的次序优化以及提供排错工具等服务^[3]。

文献[4]提出的 EPC 网络,包括 EPC 编码、Savant、对象名字服务(Object Name Service, ONS)、EPC 信息服务(Information Service)、物理标示语言(Physical Markup Language, PML)等关键技术,得到一些大学和企业的支持。EPC 组织提出 RFID 中间件的软件方面的 ALE 标准,主要包含位置与阅读器或天线的一一对应、采集数据的时间间隔、打包采集到的数据及部分报表的规范等。

Sun 公司设计的 RFID 中间件是基于文献[4]提出的 EPC 网络的基本构架,是完全端到端的解决方案,其中很重要的一点是强化了 Sun 的核心基础技术。Sun Java System RFID 软件设计支持 EPCglobal ALE 软件标准,可以过滤、集合与处理大量来自 RFID 网络边缘的数据,为企业各类应用提供流式化商务信息,可减少网络通信流量。SunJava System RFID 软件完全适用于第 2 代 EPCglobal 阅读器和标签。

Microsoft 公司提出 RFID 系统的层次结构,包括设备层、数据收集和管理层、事件和工作流管理层、基于开放标准的服务接口层和应用与解决方案层等。

IBM 公司提出一种轻型 RFID 中间件架构,主要是通过一条信息总线来代替复杂的层次结构。

Oracle 公司设计了内嵌在其应用服务器 10g 中的传感器边缘服务器,主要包含的特征有数据收集、分组、规则过滤、数据的打包发送以及在打包路由前内部数据队列的组织和管理等。

Sybase 公司推出建立在 Sybase 的相关核心产品上的 RFID 中间件,包含物理设备层、RFID 网络层、处理层、维持层、整合层、表示层和一些建模和工具组件集,除此之外还包含一些探查、警报、通告和管理的部分组件。

文献[5]是 Cambridge 大学的 Auto-ID 中心对 Real-Time RFID 的时间的性能所作研究,主要研究了在生产自动化控制领域和时间数据库的事件操作。

文献[6]是 UCLA WINMEC RFID 实验室设计的 RFID 中间件,它是在分布式架构上构建的 Web 服务,基于 XML 和 SOAP,包含了简单的捕获、平滑、过滤、路由、聚合等功能,主要应用于一些试验性的演示系统中。

文献[7]是 IBM Haifa 实验室设计了 Situation Manager,其中, Situation 本身是对复合事件概念的表现力、灵活性和可用性的扩展,它的基本内容包含一种语言和一套有效的运行时执行机制。通过结合当前系统状态和事件历史实例对事件进行收集、筛选和消耗的处理,其理论基础来自对基于规则系统的研究。

3 RFID中间件及ALE规范

中间件主要任务是对阅读器读取的标签数据进行过滤、汇集和计算,减少从阅读器传往企业应用的数据量,负责对阅读器进行管理和配置,隔离不同读写器因物理特性不同对系统管理的影响。

EPCglobal 组织在对 RFID 中间件基本功能的研究以及中

间件与上层应用接口之间的研究上提出了应用层事件(Application Level Event, ALE)规范^[8],它定义 RFID 中间件对上层应用系统应该提供的一组标准接口,以及 RFID 中间件最基本的功能:隔离,收集和过滤,中间件的位置如图 1 所示。目前大多企业研发的 RFID 中间件都遵循 ALE 规范。

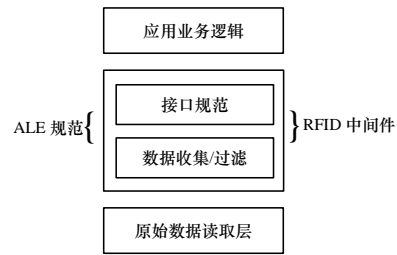


图 1 中间件的位置

在 ALE 模型中,包括以下基本概念:读周期,事件周期和报告。读周期是 RFID 中间件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。读周期的输出是 ALE 层的数据来源。事件周期可以是一个或多个读周期。它从用户角度看待读写器,可以将一个或多个读写器当作一个整体,是 ALE 接口和用户交互的最小单位。应用业务逻辑层的客户在 ALE 中定义好事件周期的边界后,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,ALE 向应用层提供数据结果^[9]。具体的 ALE 结构^[9]见图 2。

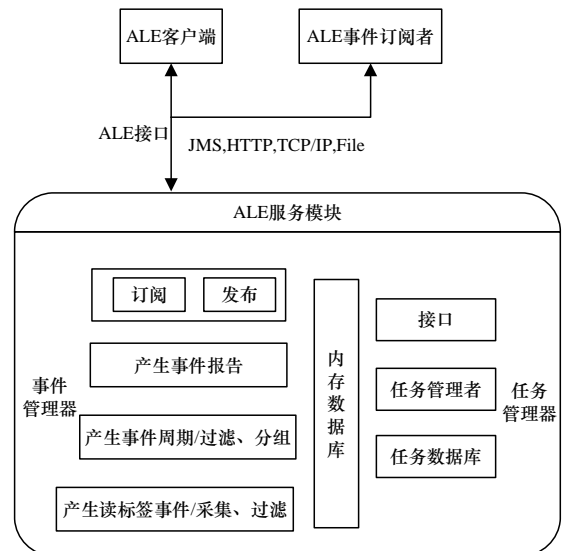


图 2 ALE 的结构

4 轻量型RFID中间件

RFID 中间件从架构上可以分为:(1)以基础架构为中心的 RFID 中间件;(2)以应用程序为中心的 RFID 中间件。

EPCglobal 设计的 ALE 软件规范是以基础架构为中心的 RFID 中间件,借助 RFID 读写器厂商提供的应用程序接口(Application Programming Interface, API),以热代码(HotCode)方式直接撰写特定读写器数据提取程序读取 Tag 数据,传送至系统应用程序或数据库,达成与后端系统或服务连接,成为以应用程序为中心的中间件。

本文研究的轻量型中间件(或称简化中间件)介于二者之间,延续 EPC 的结构,以应用程序 API 为基础,借鉴数据库集成及过滤思路,其优点为:结构简单,容易构建,系统和

思想更加实用。

4.1 轻量级中间件的构成与实现

4.1.1 系统组成和工作原理

RFID 中间件系统由 ALE 结构、API 函数等构成，包括以下特点：

(1)集成点。ALE 的特点是给出了统一的接口规范。每种读写器按其规定编写相应的适配器，以实现多类读写器读写的共享管理。从某种意义上讲，ALE 为多类读写器提供了一个集成点。

(2)临时数据库。省略 ALE 后必须为系统提供一个新的集成点。本系统采用一个 MySQL 的数据库作为新的集成点，每个 API 调用读到的 tag 数据都加入到该临时库中，由于 RFID “读的容易”的特点，作为新集成点的数据库实际上是临时数据库，其数据定时过滤和更新后，为信息系统定时提取主数据库，以及更新主数据库的内容。

(3)临时库属性选择。考虑 RFID 读头、标签的特点和主数据库对外部 TAG 数据的需求，该数据库的属性可以简单地选择为：读写器号，天线号，标签 ID，读到时间。图 3 给出了与 EPCglobal 推荐的 ALE 规范不同思路的轻量级中间件结构。

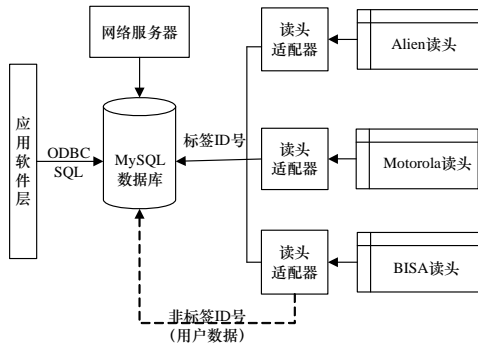


图 3 轻量级中间件结构

(4)过滤方法。主要解决由于 RFID “读的容易”的特点造成的临时库中数据的混乱和冗余、以及 RFID 不会“无中生有”（一旦读到某一标签代码、哪怕只读到一次，该标签必定出现在读写器扫描范围之内）产生的信息保留的问题。图 4 给出了基于轻量级中间件的系统结构，其中，读写器 Al 表示 Alien 的读写器；读写器 Mo 表示 Motorola 的读写器。

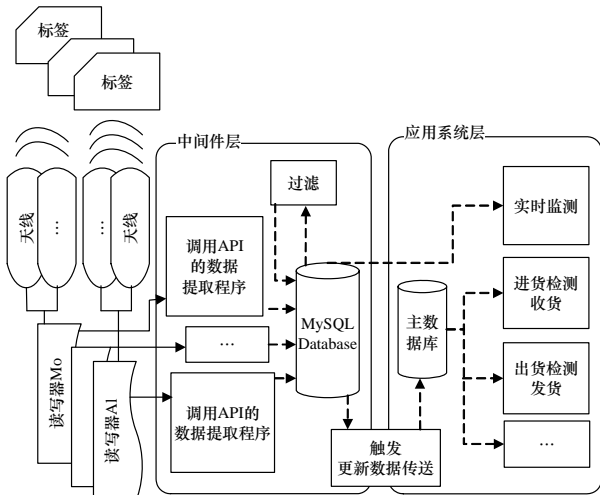


图 4 基于轻量级中间件的 RFID 系统结构

4.1.2 系统程序流程

本系统采用 MySQL+C++ 环境实现，以 Motorola XR-440 系列读写器为例说明该中间件构成、功能和程序流程^[10]。作为中间件，系统还应该提供数据转换等功能，在此不再详述。具体流程如图 5 所示。

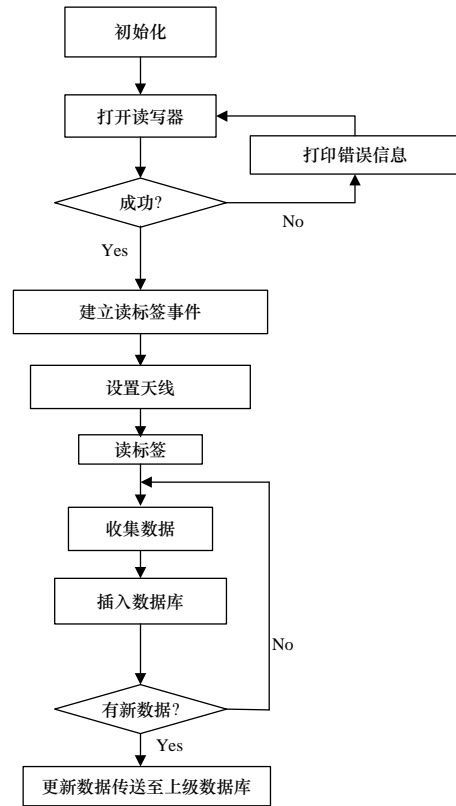


图 5 读写器读入数据处理程序流程

轻量级中间件主要进行以下操作：

(1)设置要访问的读写器，并从读写器中采集数据。通过 API 函数实现读写器与数据提取程序之间的交互，实现数据的采集。API 函数是读写器提供的 SDK 开发包中的一些函数，主要涉及以下功能的函数：读写器的打开与关闭，天线的设置，获取读写器读取的电子标签信息以及出错处理等操作。

(2)将采集到的数据放置到数据库中，并根据给定过滤算法按时进行更新。将采集到的电子标签数据显示出来并写入 MySQL 临时数据库中。通过 MySQL 提供的 API 函数，实现数据提取程序与临时数据库的对接。主要涉及连接数据库，更新数据库等功能函数的操作。

(3)集成不同种类读写器的数据，完成多协议数据的转换。由于每种读写器的开发环境不同，但最终都需要将标签数据放入到数据库中，因此这个临时数据库是这个中间件的数据集成点，每种读写器只要有相应的适配器，就可以在该轻量级中间件框架下，将数据提取到临时数据库中进行过滤、转换，为系统主数据库更新提供需要的格式。

4.2 轻量级中间件部分程序的功能说明

系统中使用的是 EPC Gen2 电子标签，根据 EPC 标准，该标签的信息位数为 96 位，因此，标签 ID 的表现形式一般为 24 位 16 进制的字符串。本系统使用 Motorola 给出的动态链接库中的函数实现数据提取程序与读写器的通信。该动态库文件设计了一些接口函数，所用到的函数主要包括以下功能：

- (1)初始化一个句柄;
- (2)为读写器配置 IP 地址和 TCP 端口;
- (3)打开一个 XR-440 读写器;
- (4)配置天线;
- (5)提取缓存在读写器中的电子标签数据;
- (6)关闭一个 XR-440 读写器。

首先进行初始化,即读入各种初始化信息,如读写器的 IP 地址、TCP 端口、天线的开或闭的状态等,然后打开读写器,即与读写器建立连接,如果能找到读写器并打开,则进行以下操作。初始化标签列表主要为标签列表分配空间并将所有变量的值都置为 0。应用动态链接库中的 API 函数对天线进行配置后,就可以读标签。读取读写器缓存的标签数据后通过数据接口函数传递,可以将这些数据信息显示在屏幕上,也可以进行入库操作。

程序实现的具体步骤如下:

(1)初始化

首先指定一个要访问的读写器 IP 地址,计算机可以通过这个 IP 地址找到读写器并与之相连。除了 IP 地址外,还有一些其他信息,如天线的开/关状态,天线的能量衰减程度设置等。这些参数可以先写入一个文件,然后在程序运行时,统一进行初始化赋值。

(2)打开读写器

打开读写器即与读写器建立连接,主要分为 3 个步骤:

1)先设定要读写的读写器 TCP/IP 参数。初始化时读取的数据要通过 API 函数传递给句柄 Reader,包括 TCP 端口和 IP 地址都要传递给 Reader,如果不成功,则显示相应的出错信息。

2)打开读写器。为读写器打开一个 API 对象,设置 TCP/IP 参数,在读写器打开后,显示已找到的读写器,并配置读写器。如果不成功,则显示出错信息。

3)介绍一下读写器的配置。根据该读写器的类型确定其所能识别的标签类型,并作相应设置。

(3)创建读标签事件

在打开读写器后,还要为读写器创建一个读取标签的事件。首先为读写器准备一个标签读取事件的名字,然后创建一个读标签事件,如果成功,则返回事件句柄,返回 0 表示创建失败。

(4)配置天线

为读数据建立一个天线的列表,并设定每个天线的初值,读取数据的程序轮询在列表中的天线,从而获得各个天线读到的标签数据。

(5)读取数据

显示上次扫描与这次扫描的时间差,读取读写器缓存中存储的数据,如果成功,则显示标签的一些信息并将其 ID 存入 Mysql 临时数据库中。

(6)显示数据

在成功读取读写器中的数据后,如果需要将数据显示在屏幕上,则分为 2 个步骤:

1)数据更新

将读到的标签 ID 存放于一个数组中,每次读到标签时都要与数组中的 ID 进行比较,如果找到相同的 ID,则更新这个标签的信息,如读到的次数,读取的时间等。如果没有找

到,则将这个新标签的 ID 写入数组中,同时写入该标签的其他信息。

2)屏幕显示

在完成更新后,可以将数据送到屏幕上:清屏,并显示本次读取的标签数据。根据需要选择显示的信息,如标签 ID 号、读写器号、天线号以及最后一次读到该 ID 的时间等内容。

(7)连接数据库与数据入库

数据提取程序与临时数据库的连接应用 MySQL 提供的 API 函数来实现。在程序中包括 MySQL 的头文件,然后通过 API 函数 mysql_connect 与 MySQL 数据库建立连接,并且将数据通过 mysql_query 与 insert 函数写入数据库中。

写入数据库中的数据包括:tagID, reader, antennaNum, lastSeen 等,其中,tagID 指标签 ID; reader 指读写器号码,即哪个读写器读到该标签; antennaNum 指天线号码,即哪个天线读到该标签; lastSeen 指最新一次读该标签的时间。

在数据写入数据库后,可以在 MySQL Query Brower 中看到标签信息,并且根据相应需要进行查询。MySQL 数据库中的数据如表 1 所示。

表 1 MySQL 数据库中的数据

编号	标签 ID	读写器号码	天线号码	最新一次读标签的时间
22	E2003411B802011207172572	01	01	2009-04-02 22:43:56
23	E2003411B802011207172564	01	01	2009-04-02 22:43:56
24	E2003411B802011207172575	01	01	2009-04-02 22:43:56
25	E2003411B802011207172565	01	01	2009-04-02 22:43:56
26	E2003411B802011207172560	01	01	2009-04-02 22:43:56

为实现中间件的功能,本系统还有相应的其他功能,本文不一一介绍。

5 结束语

本文主要研究了 RFID 轻量级中间件的结构和实现,简化了构成 EPCglobal 的 ALE 结构。该 RFID 系统结构简单清晰、原理清楚,为不同类 RFID 读写器数据的集成点提供借鉴。由于大多企业应用不会选择很多种读写器,因此企业可以按本文原理编制需要的中间件,中间件是多类 RFID 读写器工作的重要工具,相对较复杂、昂贵的中间件和售后复杂的应用维护,本文研究对 RFID 的实际应用推广具有重要意义。

参考文献

- [1] 商务部. 2005 年流通领域食品安全调查报告[Z]. (2005-11-17). http://www.gov.cn/jrzq/2005-11/15/content_99095.htm.
- [2] 孙 剑, 陈琪明. RFID 中间件在世界及中国的发展现状[J]. 物流技术与应用, 2007, 12(2): 98-101.
- [3] 丁振华, 李锦涛, 冯 波, 等. RFID 中间件研究进展[J]. 计算机工程, 2006, 32(21): 9-11.
- [4] Clark S, Traub K, Anarkat D, et al. Auto-ID Savant Specification 1.0[R]. New York, USA: Auto-ID Center, Tech. Rep.: WD-savant-1_0-20031014, 2003.
- [5] Wang Wei, McFarlane D, Brusey J. Timing Analysis of Real-time Networked RFID Systems[Z]. [2009-05-16]. http://www-ivs.cs.uni-magdeburg.de/bs/tagungen/rtn05S4_wang-RTRFID.pdf.
- [6] UCLA WinMEC RFID Laboratory. WinRFID RFID Middleware System[EB/OL]. (2005-11-22). <http://www.winmec.ucla.edu/rfid/>.

(下转第 86 页)