

ICS 35.240.50
J 07



中华人民共和国国家标准

GB/T ×××××—20××

制造过程物联网的数字化模型信息交换标准

Standard for information exchange model of Internet of Things in
Manufacturing Processes

(征求意见稿)

20××-××-××发布

20××-××-××实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会 发布

目 次

目 次.....	I
前 言.....	II
引 言.....	III
制造过程物联网的数字化模型信息交换标准.....	1
1 范围.....	1
2 规范性引用文件.....	1
3 术语与定义.....	1
4 缩略语.....	1
5 制造过程物联网的数字化模型集成需求及要求.....	2
6 制造过程物联网的体系结构.....	3
6.1 制造过程物联网软件体系结构.....	3
7 制造过程物联网的数字化模型集成架构.....	5
8 制造过程物联网的数字化模型信息交换参考模型.....	5
8.1 硬件设备与中间件系统之间的交互.....	5
8.2 应用系统与中间件系统之间的交互.....	6
9 制造过程物联网的数字化模型信息交换接口与参考实现.....	7
9.1 与硬件设备的交互接口设计.....	7
9.2 与应用系统的交互接口设计.....	8

前 言

本部分按照GB/T 1.1—2009给出的规则起草。

本标准由中国机械工业联合会提出。

本标准由全国自动化系统与集成标准化技术委员会（SAC/TC159）归口。

本标准负责起草单位：北京机械工业自动化研究所。

本标准主要起草人：。

本标准为首次发布。

引 言

制造过程物联是将网络、嵌入式、RFID、传感器等电子信息技术与制造技术相融合，实现对产品设计、制造与服务过程中制造资源与信息资源的动态感知、智能处理与优化控制的一种新型制造模式。

制造过程物联的数字化模型信息交换标准是实现制造过程物联数据互联互通的基础，可以突破现有制造系统在信息获取、控制、交互和管理方面的集成度差、协同能力弱的局限，通过对制造过程中信息交互的接口和规范的设计，有效提高制造系统的集成与互操作能力，提高制造效率和产品质量。

制造过程物联的数字化模型信息交换标准

1 范围

本标准针对目前国内外制造过程物联的技术和应用现状，参考制造过程物联功能体系结构，对制造过程物联中数字化模型的集成框架、交互模型，以及信息交互接口进行描述，并给出交互接口的参考实现及工作流程。

本标准可以为制造过程物联的实施用户、系统提供商在选型、开发及实施过程中提供相应的参考。

2 规范性引用文件

下列文件中的条款通过本部分的引用而成为本部分的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本部分，然而，鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本部分。

GB/T 18725-2008 制造业信息化 技术术语

3 术语与定义

3.1

制造资源 Manufacturing Resources

制造资源是指完成产品整个生命周期的所有生产活动的软、硬件元素，包括了设计、制造、维护等相关活动过程中涉及的所有元素。

3.2

数字化模型信息交换 Digital model for information exchange

数字化模型信息交换作为物联网的神经中枢，是连接标签读写设备和企业应用系统的纽带。它屏蔽了企业应用系统和硬件设备的关联，使硬件设备的增加，删除，更改等对企业应用系统是透明的。

3.3

多层体系结构 Multi-tier architecture

多层架构是在三层架构基础上发展起来的，将系统显式地分为几个独立的层次，实现系统之间的高内聚、低耦合性，一个层次的变化不会过多的影响其它层次的程序。

3.4

边缘处理层 Edge processing layer

边缘处理层包括：硬件设备安全接口模块、设备管理模块和数据预处理模块。与阅读设备相关联，完成与阅读设备的连接，对阅读器进行控制、管理以及保证数据传输的安全性。

3.5

事件处理层 Event processing layer

事件处理层包括：内部数据库模块、复杂事件处理模块和应用层安全服务接口模块。与前端应用系统相关联，它一方面响应企业应用系统的请求，完成与该应用系统的交互；另一方面保证了中间件与企业应用系统之间信息交互的安全性。

4 缩略语

ALE	Application Level Events	应用层事件
API	Application Program Interface	应用程序接口
CEP	Complex Event Processing	复杂事件处理
DCI	Discovery Configuration and Initialization	发现配置和初始化
DLL	Dynamic Link Library	动态链接库
DNS	Domain Name System	域名解析服务
EPC	Electronic Product Code	电子产品代码
EPCIS	EPC Information System	电子产品编码信息服务
LRRP	Low Level Reader Protocol	低级别阅读器协议
ONS	Object Naming Service	对象名解析服务
RBAC	Role-Based Access Control	基于角色的访问控制
RFID	Radio Frequency Identification	射频识别
RM	Reader Management	阅读器管理
SLRRP	Simple Low Level Reader Protocol	简单低级别阅读器协议
SOAP	Simple Object Access Protocol	简单对象访问协议
TDS	Tag Data Standards	标签数据规范
URI	Uniform Resource identifier	通用资源标识符
WMS	Warehouse Management System	仓库管理系统

5 制造过程物联网的数字化模型集成需求及要求

数字化模型信息交换作为物联网的神经中枢，是连接标签读写设备和企业应用系统的纽带。一方面，它屏蔽了企业应用系统和硬件设备的关联，使硬件设备的增加，删除，更改等对企业应用系统是透明的；另一方面，它通过自身的复杂来换取应用系统开发的简单，这大大减轻了应用系统处理庞大数据的负担。

随着我国制造业向着精益化、协同化、实时化和智能化的发展，由此引来的网络漏洞问题，引起了人们对安全问题的思考。数据交互的安全保障、传输数据的高度敏感性和确保服务质量的重要性，改善系统的兼容性，提高系统的智能化。

制造业具有以下几个特点：制造企业的产品可以用 BOM 树将构成产品的零部件明确清晰得进行描述；制造车间的每种产品都有不同的加工工艺流程，同时车间内机床的布局也没有固定的方式，工序之间的物料转移需要管理人员的宏观调度；人员密集，自动化水平低，产品的质量和生产率依赖于制造工人的技术水平；制造车间现场是物流与信息流错杂交汇的场所，生产状况繁杂，不易掌控。

针对制造车间的这些特点以及车间内现在存在的问题，需要利用车间智能制造对象对车间内每个制造单元的局部优化，从而达到对车间整体生产流程优化的目的，改变传统的车间的生产模式，实时控制和管理从车间作业计划的下达到最终产品完工检验入库全过程的实时数据采集、数据流程控制、数据统计分析等，实现对制造车间数据的全面管理，合理优化使用采集到的实时数据，提高数据处理的效率，从而提升车间生产能力，保证产品的交货期，系统功能的需求分析如下：

a) 高效的制造车间数据采集方法

在制造车间现场制造数据的采集一直是一个难题，传统的以纸质文档作为载体、手工记录的数据采集方式存在着很明显的缺陷，纸质文档容易遗失、损坏，人工记录难免出错，数据查找起来困难；条形码技术又因为其识别距离很短，读取的方向性要求高，读取速度慢，能携带的数据量小且不可重写，易污损等缺点，限制了其在离散制造车间的广泛使用；像机械打标、激光刻字以及喷码技术都曾用在离散制造车间进行在制品的标识，但都因为各自的局限性不能在车间进行

全面的使用。射频识别技术由于其特有的非接触远距离识别、电子标签信息存储量大等优点，特别适合对离散制造车间的制造资源和在制品进行自动标识，实现对离散制造车间现场制造数据的实时采集。

b) 有效的制造车间生产过程控制方法

对于车间的每个生产工位，在对在制品进行加工的过程中，通过可视化技术第一时间向操作工人发放所需要的工艺文件、零件图纸、数控程序以及检验标准等，同时将机床运转的状况、工装配备状况以及物料的准备状况进行实时得反馈，指导操作工人进行每道工序的生产加工，当出现机床故障、物料不足的情况下，及时进行工位的报警，提醒现场操作工人进行及时处理，避免质量隐患，从而提高制造过程的生产效率和加工质量。而对于离散制造车间的管理人员，可以实时的掌握物料从投入到产品入库全过程的制造数据，包括生产设备的信息和状态、生产的工人、采用的工装、每道工序的质检数据和时间等，使得管理人员能够针对目前的生产状况做出正确的生产决策。

c) 对车间的基础数据和采集的实时数据进行统一管理

需要建立一个统一的数据模型全面统筹整个制造车间所需要的基础数据和现场采集到的数据，基础数据是指车间所有制造资源的属性数据和生产制造过程中涉及到文件，包括机床、人员、工装等的基本信息以及工位生产计划、工艺文件、质检文件、零件图纸等等，它是各种编码信息转化为可以理解的实际意义的基础，同时，将现场采集到的半结构化的数据与结构化的基础数据相互融合进行统一的数据建模，使管理人员进行对车间的资源状况有全局的认识，为物联网车间数据管理系统的各项功能的实现提供全面可靠的实时数据支持。

d) 全面的离散制造车间生产数据统计分析

传统的车间生产数据的统计分析，由于现场制造数据来源的实时性和准确性无法保障，导致数据的统计分析不准确，对车间生产制造活动的指导具有明显的滞后性，通过智能制造对象技术能获取的丰富的全面的离散制造车间的实时数据，基于这些准确的制造数据进行统计分析，使得管理人员可以准确清晰地获得在制品、机床、工人的状况，轻松得掌握每道工序的工时，对整个产品以及订单的进度现状以及预测可以做出科学的判断和决策，合理得安排车间的制造资源，优化资源配置，减少不必要的停机时间，达到对车间资源最大化的使用，提高生产效率。

6 制造过程物联的体系结构

6.1 制造过程物联软件体系结构

制造过程物联是体现先进传感技术和管理理念、用先进软/硬件实现技术、支持制造过程优化管理的应用软件系统。制造过程物联包括四个主要功能体系，即制造资源物物互联体系、动态事件实时感知体系、信息整合体系和实时信息驱动的应用服务体系。

制造过程物联的软件体系结构是制造过程物联系统实现过程中需要遵循的指导性框架规范，具体到制造过程物联系统体系结构的实现技术，可采用的有浏览器/服务器(B/S)结构、客户机/服务器(C/S)结构，多层体系结构，面向服务的构架等。

a) 多层体系结构

制造过程物联体系结构明确要求采用基于软构件的多层架构。所谓多层架构是在三层架构基础上发展起来的。三层架构将软件系统分成表示层、逻辑层及数据层。其中表示层负责实现客户端的用户界面，将从逻辑层获得的数据展现给用户，并将从客户端获得的数据提供给逻辑层进行处理。逻辑层负责实现真正的业务逻辑处理，将处理的结果一方面提供给表示层，另一方面发送到数据层进行存储。数据层主要负责数据的存储及查询。这种架构将业务逻辑与表示界面分开，有利于实现软件的瘦客户端，应用的快速重组，便于系统的维护和升级。而软件的多层架构则主要是将逻辑层分成更多的层次，例如应用服务器中间件、数据库驱动。图2给出了三层架构的参考模型。

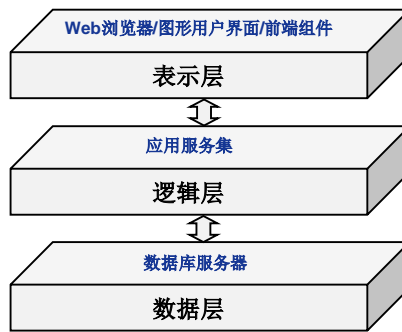


图2 三层架构

制造过程物联系统采用多层架构主要是为了提高系统的可扩展性。由于将系统显式地分为几个独立的层次，而各个层次如果按照面向对象的原则能够实现高内聚、低耦合性，则一个层次的变化就不会过多的影响其它层次的程序。同样逻辑层与数据层的隔离可以使制造过程物联系统在数据层采用不同的解决方案，例如多数据库系统的适应性。

多层架构的具体实现形式有很多种，例如目前十分流行的客户机/服务器(C/S)结构和浏览器/服务器(B/S)结构。

图3给出了多层体系架构的组成，客户机/服务器(C/S)结构还是浏览器/服务器(B/S)结构的软件都可以实现多层架构。

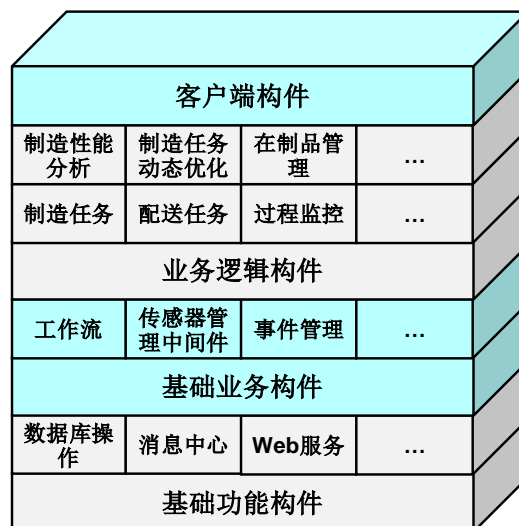


图3 多层架构

b) 基于构件的体系结构

构件的作用是构造新的软件系统可以不必每次从零做起，直接使用已有的构件，即可组装（或加以合理修改）成新的系统。

制造过程物联体系结构强调在逻辑层应采用基于面向对象的软构件技术。面向对象的软构件实际上是“对象+标准的接口”。而现行的接口标准主要有COM+、JavaBean(EJB)、Web服务等。采用软构件技术的主要目的在于使系统可以支持基于工作流的、面向业务流程的重组，从而提高制造过程物联系统的可配置性和适应性。制造过程物联系统推荐将构件根据其业务类型划分为基础功能构件、基础业务构件、业务逻辑构件、客户端构件4个层次。基础功能构件负责对操作系统、数据库、应用服务器、Web服务器等的操作或通讯。基础业务构件是系统中的通用构件，它们被具体的制造过程物联业务处理构件调用。业务逻辑构件是实现具体的制造过程物联业务逻辑的构件，每一个构件可以完成某一方面的业务处理功能，例如过程监控业务构件、性能分析业务构件等。客户端构件是用来完成客户端界面的构件，包括动态页面程序，自定义标签，客户端控件等。

7 制造过程物联网的数字化模型集成架构

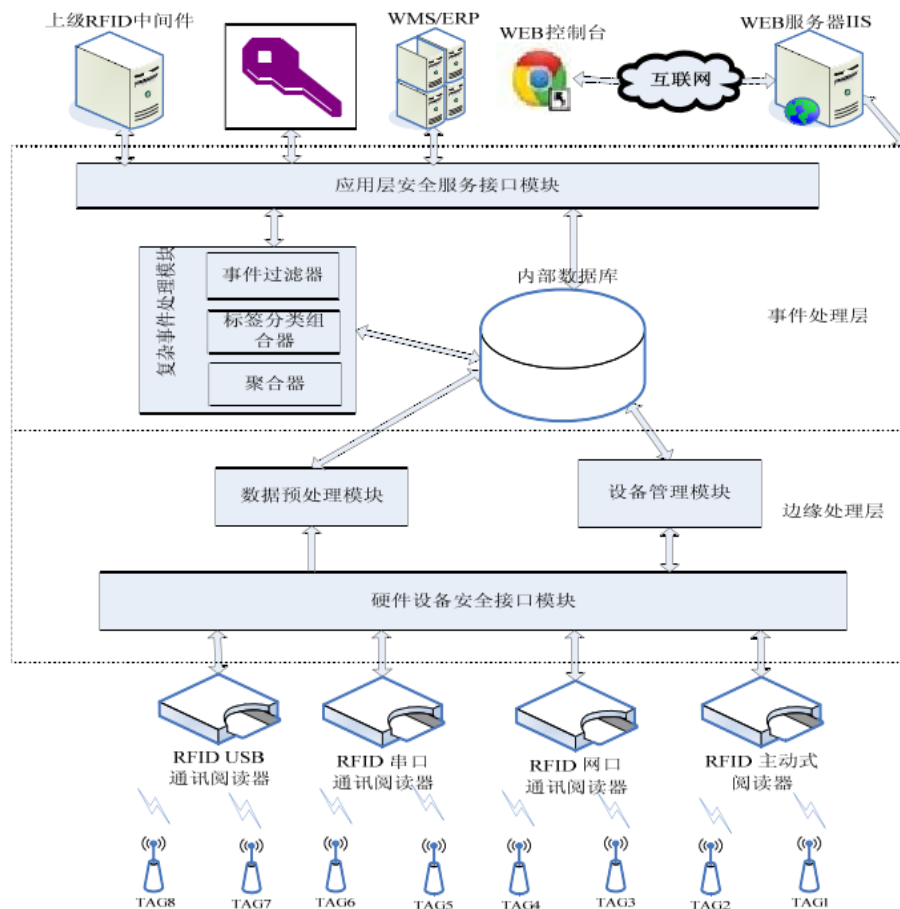


图4 制造物联数字化模型总体结构图

如图4所示，制造过程物联网的数字化模型是由边缘处理层和事件处理层组成的，边缘处理层包括：硬件设备安全接口模块、设备管理模块和数据预处理模块。事件处理层包括：内部数据库模块、复杂事件处理模块和应用层安全服务接口模块。边缘处理层与阅读设备相关联，完成与阅读设备的连接，对阅读器进行控制、管理以及保证数据传输的安全性。事件处理层与前端应用系统相关联，它一方面响应企业应用系统的请求，完成与该应用系统的交互；另一方面保证了中间件与企业应用系统之间信息交互的安全性。

8 制造过程物联网的数字化模型信息交换参考模型

8.1 硬件设备与中间件系统之间的交互

中间件系统提供两个接口和外界进行交互分别是：硬件设备安全接口、应用层安全服务接口。中间件系统和它们之间交互的信息包括三类：对阅读器进行操作和管理的控制命令；原始标签数据和事件；应用程序客户端向中间件系统发送的请求消息。

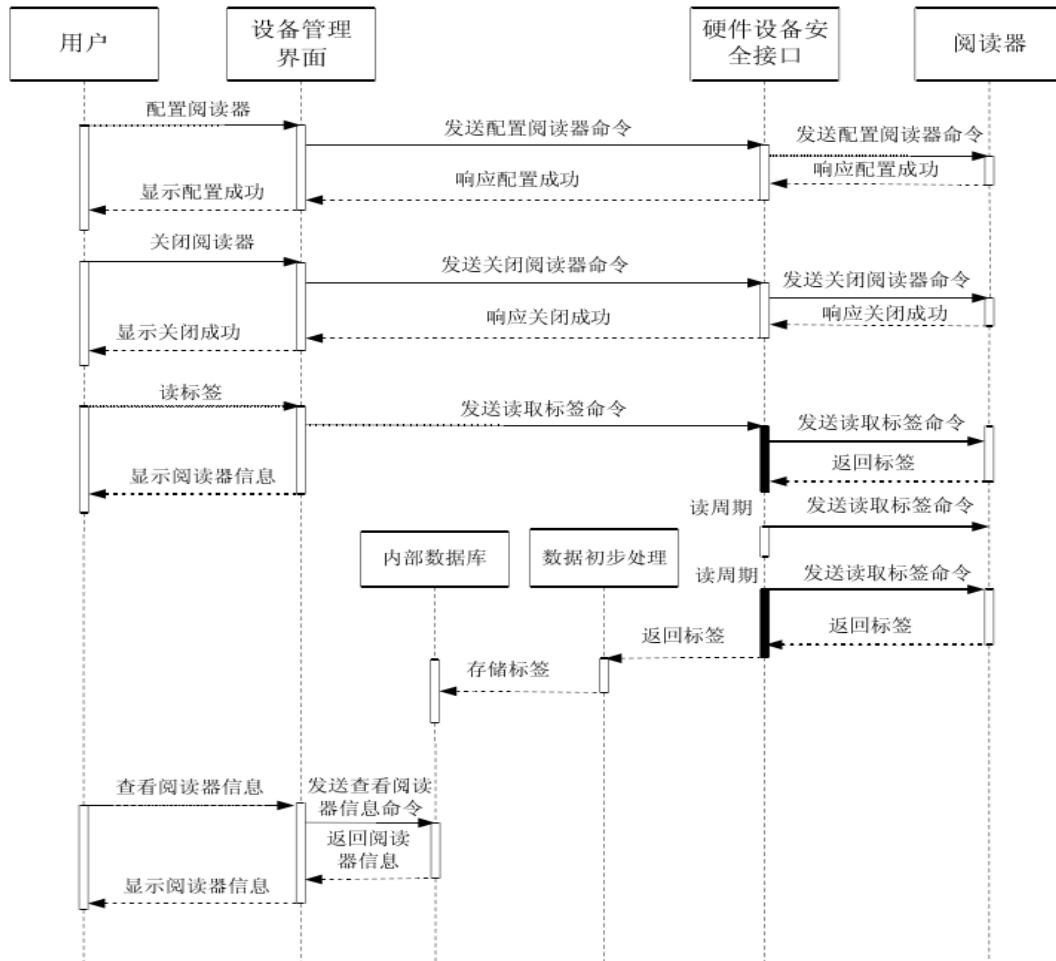


图 5 硬件设备与中间件系统之间的交互

图 5 中用序列的形式清晰明了的介绍了硬件设备与中间件系统之间的交互。中间件系统按读周期向阅读器发送读命令，阅读器对在阅读范围内的原始标签进行采集，然后将这些数据传送给中间件。当数据过滤时间到后，开始数据的预处理，最后将统一的标签格式存储到内部数据库中。对阅读器的控制和管理工作是用户通过浏览器对界面进行操作完成的。其操作的内容包括：配置阅读器、开启和关闭阅读器以及查看阅读器信息等。

8.2 应用系统与中间件系统之间的交互

图 6 中所示的处理序列介绍了应用系统与中间件系统之间的交互，该交互过程假设为一个正常的调用场景，没有错误出现。企业应用系统客户端向中间件系统发送请求，定义一个名为 Name1 的事件周期，应用层服务安全接口解析出事件周期 ECSpec 里的信息后存入内部数据库。接着客户端再发送请求 poll (Name1) 获取事件 Name1 的报告，中间件收到请求后，通过接口解析并调用相应的复杂事件处理模块中的子模块。它根据 ECSpec 定义的边界值和逻辑阅读器从数据库中提取需要的标签数据，若读取到了数据，则按 ECSpec 定义的规则进行事件处理，最后将处理后的事件按照定义的要求生成事件报告 ECRReport(XML 格式)，通过接口发送给客户端，若没有读取到数据，则返回一个空的 ECRReport。

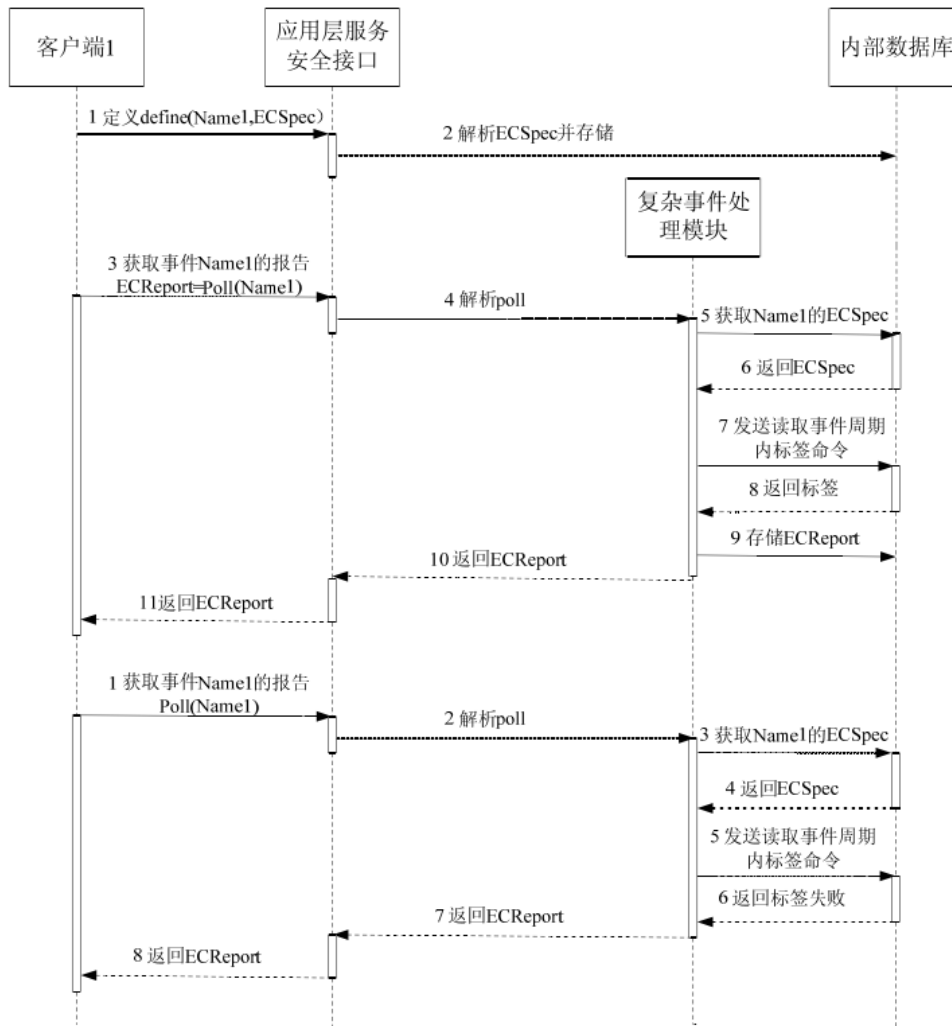


图 6 应用系统与中间件之间的交互

9 制造过程物联网的数字化模型信息交换接口与参考实现

9.1 与硬件设备的交互接口设计

支持网口、串口和 USB 连接方式，有利于系统产品级的扩展和升级。按读周期采集阅读器传送来的标签数据，针对不同的阅读器，选择其对应的控制器进行驱动和控制。

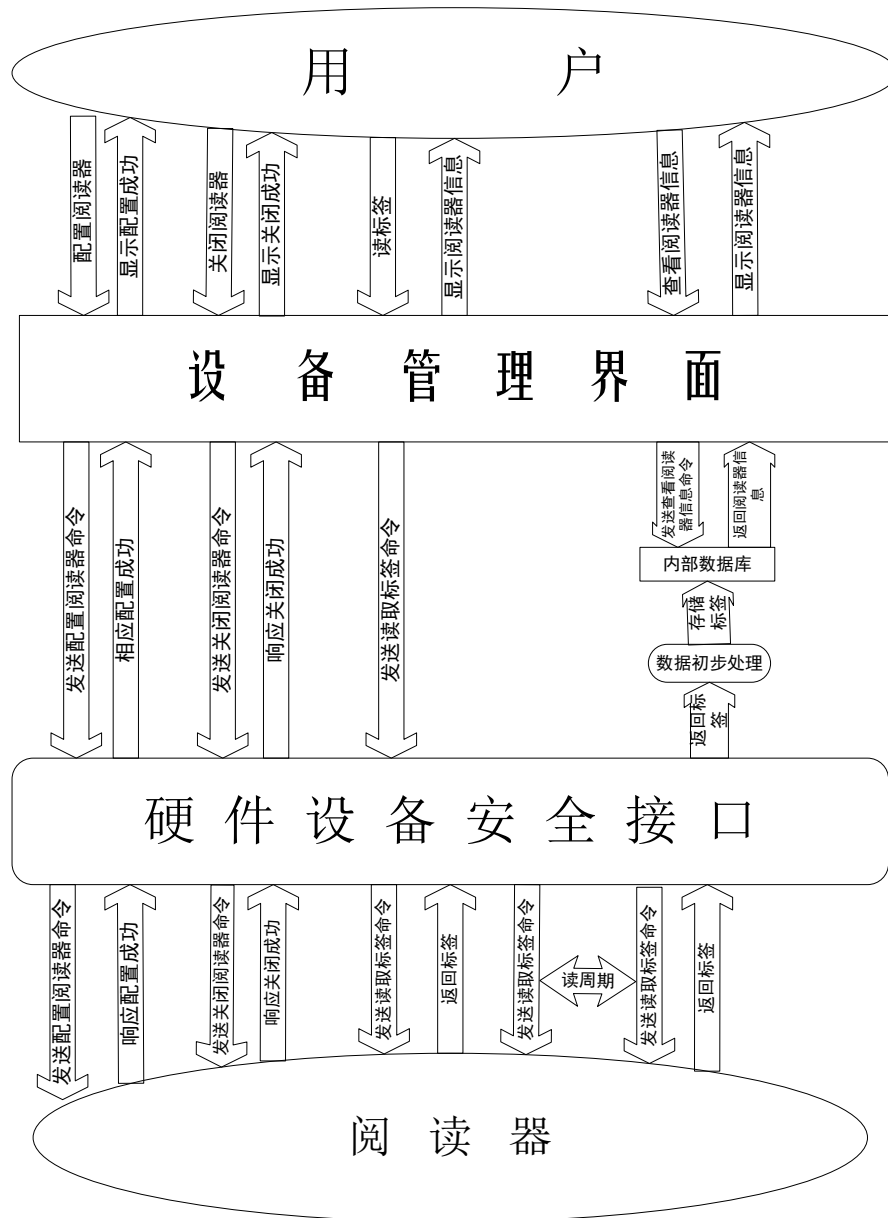


图7 与硬件设备的交互接口设计

如图7所示，该模块可兼容不同类型的通信协议如：RS-232、USB2.0、TCP和UDP等。我们根据阅读器厂商提供的API函数编写控制程序。

对控制程序的封装有两种形式：一种是定义一个标准控制器接口，再根据不同类型的阅读设备提供的开发包编写类，该类继承自标准控制器接口，并逐一实现接口中的功能函数，从而实现中间件对各种阅读设备进行统一管理；另一种方式是将提供的API函数分模块封装成动态链接库形式，这样不仅提高了软件的复用性，也大大增强了中间件系统的灵活性和扩展性。我们采用后者实现中间件与硬件设备之间的接口。

9.2 与应用系统的交互接口设计

9.2.1 ALE接口

借鉴ALE规范定义的接口并对其进行扩展，实现中间件系统与前端应用系统和上级RFID中间件系统之间进行通信的接口。该部分根据系统设计思路，采用同步Poll模式和Immediate模式提供六种服务，具体服务描述如下：

定义事件周期服务：定义一个ECSpec，其名称为参数spec_Name，定义的内容

包含在参数 spec 中。该服务方法声明如下：

[WebMethod]

```
public void define(string spec_Name,ECSpec spec)
```

数据结构 ECSpec 包含了这个事件周期的边界值定义和事件周期报告的格式，事件过滤和组合条件等。

具体实现流程：解析事件周期 ECSpec，并存入数据库的 ECSpec 表和 ECReportSpec 表中。

同步抽取数据服务：请求一个名称为 spec_Name 的事件周期，同步返回一个报告 ECReport。Web 服务方法声明如下：

[WebMethod]

```
public ECReports poll(string spec_Name)
```

该方法中调用的主要方法有：Aggregate()，TimeFilter()，ProductFilter()，

TagIDFilter()，SortGroup()。这些方法在复杂事件处理模块中已经定义。

具体实现流程：根据名称 spec_Name 查询数据库的 ECSpec 表，找到对应的 spec 并根据该定义的数据处理规则来调用复杂事件处理模块中定义的方法进行数据的过滤、分类和聚合，数据处理完成后，按照定义的事件周期报告格式生成 ECReport，并返回给订阅者。立即获取数据服务：直接定义 spec 并同步返回报告 ECReports。Web 服务方法声明如下：

[WebMethod]

```
public ECReports immediate(ECSpec spec)
```

具体实现流程：先根据 spec 创建一个未命名的事件周期 ECSpec，接着根据定义的数据处理规则调用复杂事件处理模块中定义的方法进行数据处理，处理完后立即返回事件报告 ECReports。

该方法和 poll()方法的本质区别是需要直接定义事件周期 spec。获取定义的事件周期服务：获取名称为 spec_Name 的事件周期定义。Web 服务方法声明如下：

[WebMethod]

```
public ECSpec getECSpec(string spec_Name)
```

具体实现流程：查询数据库 ECSpec 表、ECReportSpec 表，获取名称为 sepc_Name 的记录。对 ALE 规范进行扩展的 ECSpec 相关类图如图 4-11 所示，该图只展示了扩展的部分，未扩展的部分没有表示出来。中间件系统不需要一直监听客户端，因此为了增强客户端收集数据的实时性，

需要对 ECBoundarySpec 对象进行扩展。该对象增加了两个成员：startTime 和 endTime，这两个成员用来表示客户端需要的是该段时间里出现的标签数据，并且开始的时间不能超过当前时间。根据典型环境应用的需求，在 ECReportSpec 对象中扩展了时间过滤规则定义成员 ECTimeFilterSpec 对象。该对象中的成员：startFilterTime 和 endFilterTime 表示过滤的时间间隔，两个值必须分别小于 startTime 和 endTime。

9.2.2 外部数据库接口

为了以后扩展需要，该系统为客户端提供访问外部主流数据库的服务接口：事件查询服务和事件添加服务。ADO.NET 提供了一组封装好的类，用于操作不同的数据库。命名空间 System.Data.SqlClient 包含连接 SQL Server 数据库的类；命名空间 System.Data.OracleClient 包含连接 Oracle 数据库的类，命名空间 System.Data.Odbc 包含连接 Access 数据库及其他 ODBC 数据库的类。