

# 基于万维网服务的设备集成技术

金路<sup>1</sup>, 沈苏彬<sup>2</sup>

(1. 南京邮电大学 物联网学院, 江苏 南京 210003; 2. 南京邮电大学 计算机学院, 江苏 南京 210046)

**摘要:** 通过对万维网服务集成技术的研究, 提出使用 DPWS (万维网服务设备简本) 来解决各类设备的集成问题。依托网络摄像机集成的实际需求, 归纳设备集成过程中的难点。使用 DPWS 实现了基于万维网服务的应用接口, 可集成动态发现、事件等特性, 达到了迅速集成联网设备的目的。测试结果表明该方案具有较高的性能、伸缩性、顽健性和普适性。

**关键词:** 面向服务的设备体系; 面向服务体系; 万维网服务; 万维网服务的设备简本

中图分类号: TP391

文献标识码: A

文章编号: 1000-436X(2012)Z2-0244-06

## Device integration technology based on Web services

JIN Lu<sup>1</sup>, SHEN Su-bin<sup>2</sup>

(1. School of IoT, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;

2. School of Computer/Software, Nanjing University of Posts and Telecommunications, Nanjing 210046, China)

**Abstract:** Through studied on integration technologies based on the web services, DPWS (devices profile for Web service) was applied to solve the integration issues of different kinds of devices. Relying on the requirements of integrating networked cameras, the difficulties in the integration process of embedded devices were summarized. Application interfaces based on web services was implemented by using DPWS and the features like dynamic discovery, eventing were integrated. The test results show that the system has high performance, scalability, robustness and universality.

**Key words:** service-oriented device architecture; service-oriented architectures; Web services; devices profile for Web service

### 1 引言

传统嵌入式设备中的应用通常是独立存在或以专用协议方式提供自身信息与服务, 这就导致相关系统搭建时需要处理的内容多、参数意义不明确、代码构建异常复杂。因此这些设备仅能同指定管理系统互连, 不同种类管理平台和设备之间不具备互连、互通的能力。

万维网服务 (Web 服务) 是一种支持网络设备间互操作的软件体系<sup>[1]</sup>, 目前主要应用于大型的企业信息系统中。Web 服务是解决设备间互连、互通的最佳实践, 且随着嵌入式设备性能的快速提升,

原本高层次的 Web 服务已具备在低层次的嵌入式设备中实现的条件。而 Web 服务技术的组件化思想, 将使得数据表示、相关系统编程的难度大大降低, 且具备跨平台、跨语言能力。

本文通过研究基于万维网服务的集成技术, 提出使用 DPWS<sup>[2]</sup> (万维网服务的设备简本) 来解决嵌入式设备的集成问题。依托网络摄像机集成的实际需求, 归纳嵌入式设备集成过程中的难点。基于 DPWS 实现了摄像机系统集成, 使摄像机提供基于万维网服务的应用接口, 且具备动态发现与感知、事件通报、服务调用等功能, 并分析了系统的性能、可伸缩性、顽健性和普适性。

收稿日期: 2012-10-23

项目基金: 江苏省科技支撑计划基金资助项目 (BE2009157)

Foundation Item: The Scientific and Technological Funding Projects of Jiangsu Province (BE2009157)

## 2 DPWS 及其相关技术

DPWS 最初是由微软和一些网络打印机设备商提出来的、关于网络设备的一个万维网服务简本，简本说明了不同场景下的使用规范，目前已在 OASIS（结构信息标准化推进组织）得到标准化。DPWS 是基于 WS-\*标准的，充分利用现有 Web 技术，包括 HTTP（超文本传送协议）、SOAP（简单对象访问协议）和 XML（可扩展标记语言）技术，因此，它真正与平台无关。实现 DPWS 的协议栈如图 1 所示。

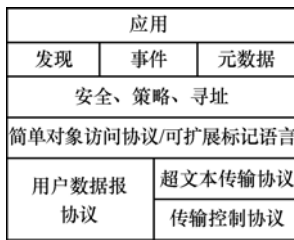


图 1 实现 DPWS 的协议栈

**WS-addressing:** WS-addressing<sup>[3]</sup> (WS-寻址) 通过引入“端点引用”(EPR)和“消息信息头”(MI) 2 个概念来解决 Web 服务寻址机制以及消息传输的问题。

**WS-discovery:** 自动发现是客户端和设备消息交互的第一步，主要通过 WS-discovery<sup>[4]</sup> (WS-发现) 协议实现。WS-discovery 通过设备类型 (type) 和设备所在的范围 (scope) 来确定和登录设备服务。WS-discovery 不仅支持设备的自动发现，还支持发现代理。设备可以直接跟代理通信，而不用给网络多播或广播消息。

**WS-eventing:** WS-eventing<sup>[5]</sup> (WS-事件) 定义了一个用于管理 Web 服务订阅的协议，它基于事件机制。该协议定义了 3 个角色：订阅者、事件源和订阅管理器。订阅者可以向事件源订阅事件，事件源在事件发生时会将该事件的相关数据发送给这个订阅者。订阅请求数据分组中包含事件的传送方式及事件的过滤信息。订阅管理器负责管理订阅者的订阅请求，如到期时间、订阅事件等。

**WS-metadata exchange:** WS-metadata exchange<sup>[6]</sup> (WS-元数据交换) 的是一种规范，它定义了一组数据类型和操作用来检索服务端点的元数据。元数据是一组描述端点基本信息的静态数据，这个基本信息是其他端点打算与此端点进行交互的必要前提。

**WS-security:** WS-security<sup>[7]</sup> (WS-安全) 为 SOAP

消息提供消息完整性和保密性的机制。规范定义了如何使用不同安全令牌格式 (如 SAML、Kerberos、X.509) 对部分或全部的 SOAP 消息进行签名或者加密，它基于现有 XML Signature 协议和 XML Encryption 协议。

DPWS 使用的一组 WS 标准带来了如安全的万维网服务、动态发现、万维网服务描述、网络上订阅或接收事件消息等多种特性。

## 3 系统设计

系统旨在研究并实现基于 DPWS 的设备集成方法，使其提供基于万维网服务的应用接口，且具备与其他设备、管理控制平台间对接交互的功能，并提供如发现、通知等特性。

目前有如 WS4D<sup>[8]</sup> (面向设备的万维网服务)、SOA4D (面向设备服务体系) 和 SOCRADES (面向服务的跨层分布式智能嵌入式设备基础设施) 提供的基于 DPWS 的开源实现，微软也围绕 DPWS 提供 WSDAPI 接口。

WS4D 是欧洲 ITEA<sup>[9]</sup> 项目 SIRENA<sup>[10]</sup> (实时嵌入式网络应用服务基础设施) 中的子项目。而 WS4D-gSOAP 是 WS4D 中基于 gSOAP 的工具包，它遵从 DPWS。由于 gSOAP 是一个高效的基于 C/C++ 语言的 Web 服务开发平台，因此适用于小型的资源受限设备，系统采用 WS4D-gSOAP 开源工具包实现 DPWS 的功能。

本文的实验平台是一群连接于局域网遗留的一群网络摄像机系统，其结构如图 2 所示。

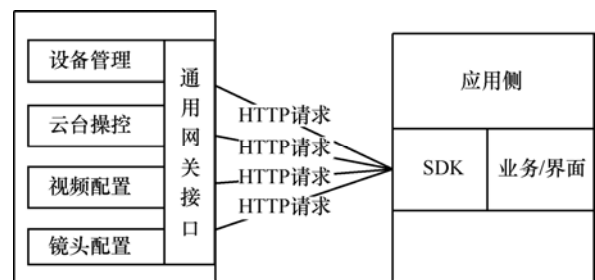


图 2 原有系统集成前结构

原系统使用基于 CGI (通用网关接口) 提供设备应用接口。例如，对于配置摄像机网络接口参数的操作，原有的嵌入式 Web 控制平台和 C/S 结构控制平台调用如下格式的 HTTP 请求：

`http://Ip/cgi-bin/LanConfigure.cgi?Action&IpAddress&Netmask&dns&Gateway&UserName`

浏览器收到此请求后, 将传入的各参数设置成系统的环境变量作为数据收发的通道, CGI 程序通过标准的输入输出, 读取和反馈数据。

原有的网络摄像机具有四大模块功能模块(如表 1 所示), 分别是设备管理、云台操控、视频配置和镜头配置。

表 1 原有系统中设备服务模块

模块名称	功能
设备管理	网络配置、系统配置、安全管理配置、用户及权限管理、软件更新等
云台操控	云台转动、雨刷、自动归位、线扫功能、预置点、轨迹
视频配置	视频流获取与视频分辨率、码流设置
镜头配置	聚焦、视角、镜头状态

在应用程序端, 基于标准 C 的 SDK(软件开发套件)抽象了四大模块的功能接口, 使得应用程序直接调用 CGI 接口, 即可完成相关功能, 透明化处理 HTTP 请求。因此, 在 UI(用户接口)层看来, 对摄像机的操作仿佛不存在网络通信过程, 而是直接由 SDK 提供的。

很明显, 这样的服务接口应用有如下缺点:

- 1) 不具备设备寻址与发现功能, 必须得知摄像机的 IP 地址后才能发起请求;
- 2) 安全性低, 可以通过网络数据分组的分析从而使得任何人具备使用服务的能力;
- 3) 可操作性差, 服务的使用必须借助于文档, 而文档需要说明每个请求/响应包含字段的含义;
- 4) 语义不明确导致开发难度大;
- 5) 没有事件机制, 较为先进的摄像机一般提供报警或“状态—动作—通知”的能力, 即摄像机达到某状态后, 根据预先设定可自主激活某动作并通知用户。

集成后的系统, 希望具有平台独立、网络结构简单、顽健性强、可大规模部署等特性, 同时能够克服上述缺点。按照以上需求, 针对一般的设备集成过程, 可以归纳出如下 5 个难点: 寻址、发现、描述、控制和事件。

- 1) 寻址。具备网络交互的基础。对于 IP 网络, 寻址能力通过 IP 协议提供。
- 2) 发现。寻找合适的设备。设备在开放的网络环境下, 需要让其他实体发现彼此。当一个设备添加到网络后, 发现协议可使设备将其服务暴露在网络中。通常情况下, 搜索和暴露端点使用多播消息请求、单播响应来实现。

3) 描述。设备描述自身能力的方法。例如, 当控制端发现了一个受控设备, 控制端必须检索受控设备的相关描述, 这样的描述一般称为“元数据”。设备元数据可能包括制造商的名称、版本、序列号、型号描述和服务描述等。设备描述同时定义请求和响应的动作及相关的消息格式。

4) 控制。利用设备的服务功能。当设备互相感知并得到相关能力的描述信息后, 控制端设备可使用设备的服务控制该受控设备。要调用一个设备的服务, 控制端发送符合数据格式的控制消息, 并处理返回的消息。控制过程中使用的服务是相关设备根据自身应用场景而定制的, 不同的设备可能有不同的服务。

5) 事件。网络设备的事件通报机制。设备事件通知主要是侦测设备自身异常事件、状态通报等。一般使用“发布—订阅”机制来实现。订阅指订阅者申请某事件的过程。订阅者须向发布者发送一个订阅某事件的请求。当订阅者不再关注这类事件时, 订阅须向发布者发送一个取消订阅请求。

系统使用 DPWS 使得上述难题迎刃而解。其对原系统的改造过程需要进行如下 4 步工作才能将现有的 HTTP/CGI 服务提供方式转换成基于 DPWS 的服务方式。

- 1) 将基应用侧的 SDK 移植到嵌入式环境;
- 2) 编写万维网服务接口的描述, 形成 WSDL 文档;
- 3) 编写数据语义转换的适配器;
- 4) 利用 WS4D-gSOAP 提供关于 DPWS 中各协议栈对应的实现。

集成后, 摄像机与应用程序间应形成如图 3 所示的结构。

## 4 系统实现

### 4.1 移植 SDK

系统中原有 SDK 基于 windows 平台, 它提供对摄像机系统主要模块的方法调用, 且使用了标准 C 语言。经过 SDK, 透明化处理 HTTP 请求。也就是说, 只要调用了 SDK 的某个方法, 提供相关参数, 就能获得相应摄像机的服务功能。移植 SDK 最主要的工作是使原来基于 winsock 的网络通信修改成具备在摄像机嵌入式 Linux 下通信的能力。可通过定义 Win32 或 Linux 等条件编译开关控制不同函数参与编译, 以屏蔽平台接口函数的差异。移植过后, 由于 SDK 本身内置于摄像机, SDK 用回环地址调用

原 HTTP 服务。显然本地 HTTP 调用的方式可能影响设备的性能，SDK 也可以不用 HTTP 调用，而改用本地调用的方式直接调用四大模块的功能。

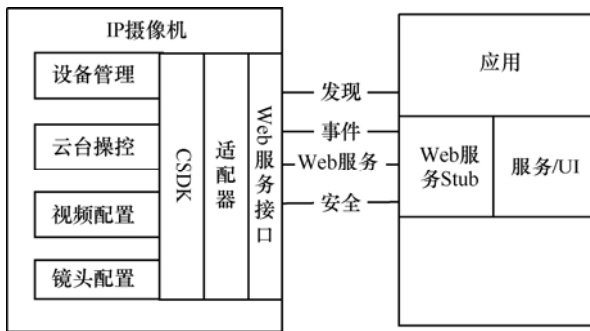


图 3 系统集成后结构

### 4.2 编写万维网服务接口

万维网服务接口是 SDK 接口更高层的抽象，它用 WSDL（万维网服务描述语言）来描述，定义了行为准则和数据格式，最终形成 WSDL 文档。详细的制定方针可参照领域内的标准，为了节省开发时间，设备的 WSDL 文档借用网络视频管理领域的标准 ONVIF（开放网络视频接口论坛）提供的 WSDL 定义。

### 4.3 编写适配器

例如，对于获取网络信息，其返回值如表 2 所示。

表 2 获取网络信息的返回值

返回值	含义
MAC address	MAC 地址
IP address	IP 地址
net mask	子网掩码
gateway	网关
DNS	DNS

适配器在调用了 SDK 下 GetNetworkInfo 函数后，其返回上述格式的数据。而在万维网服务接口中，这些数据对应 NetworkInfoResponse 这个类型。因此需要实例化一个这样的对象，并将 SDK 请求后的返回值拆分并识别，最终将对应的数据值赋予相应的字段。适配器实际上起数据分析并转换的工作。

### 4.4 提供 DPWS 协议栈的功能

在提供 DPWS 功能前，系统首先需要选择具体的互连策略，因为 DPWS 支持“中心注册代理”和“对等”2 种发现模式，分别对应代理和直连 2 种互连策略。不同的互连策略对系统有着不同的影响。

考虑到网络摄像机系统性能良好，系统采用直连的互连策略。直连策略是指将服务程序集成到网

络设备中，它使得设备自身具备提供所有业务的能力，同时简化了网络的复杂性。直连机制对设备能力的要求相对较高。系统中凡是涉及报文交互的两端都是直接交互的。

WS4D-gSOAP 与 gSOAP 使用了类似的开发流程。该工具包生成后的代码遵从 DPWS，这样开发者只需要将工作重心集中在业务代码的编写上。

**定义 1** 设备元数据。元数据的定义包括 3 个部分：Relationship、ThisModel、ThisDevice。描述了该设备的基本信息，如型号、生产厂家、服务类型、名称等。这些信息以固定的格式保存在一个 XML 文档中。

构建服务端与客户端。工具包使用元数据和服务定义构建.gsoap 文件，该文件由 gSOAP 生成。之后 WS4D-gSOAP 可根据这个文件生成客户端和服务端的代码。CMake 作为代码和文档的生成器，不仅根据实际功能需要编写.cmake 配置文件，还需要用户填写基本的配置信息，如指定编译器、gSOAP 的安装路径、WS4D-gSOAP 安装路径、项目生成路径、IPV6 支持等。当 CMake 配置正确，并且生成成功时，可以获得项目开发相关源码。这些源码基于用户声明的万维网服务接口和相关元数据，且遵从 DPWS。

服务端实现主要包括以下步骤：初始化设备与服务，设置元数据信息并更新，设备广播运行中的服务，处理接收到的请求消息，实现提供服务的业务逻辑。客户端实现包括下面的步骤：初始化 DPWS 与 SOAP 句柄、发现端点、调用服务。客户端和设备端的交互序列如图 4 所示。

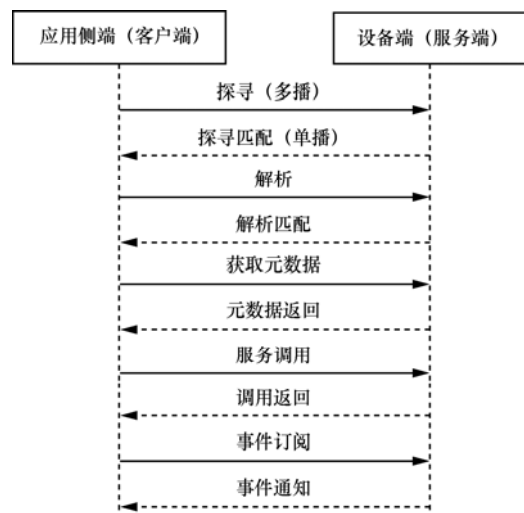


图 4 系统集成后结构

上述实现仅需要使用工具生成代码，并按照工具说明书，按流程调用相关函数即可实现兼容 DPWS 协议的服务端和客户端。

### 5 系统测试和分析

通过 3 种不同软件测试 ONVIF 提供的摄像机云台 PTZ（平移/倾斜/缩放）功能，测试项目包括设备服务的发现与调用。测试中使用的 3 种软件分别为：基于 WS4D-gSOAP 编写的客户端程序、基于 WS4D-JMEDS 编写的 DPWS Explorer、基于 WCF 编写的网络视频管理系统软件。

1) 运行设备服务程序，服务程序等待多播的“探寻”报文。

2) 运行客户端程序，客户端程序发现服务端点后，主动调用设备提供的服务。

3) 运行“java-jar ws4d-explorer.jar”命令启动 DPWS Explorer，发现设备提供的服务。

前 3 步显然已经达到了设备集成的目的，因为此刻客户端程序能够发现所有该类型的设备，并调用其服务。再次，由于设备启动时会主动发送“WS-发现”的“Hello”报文，告知其他设备和客户端有关自身的存在，其他设备或客户端也具备调用其服务的能力。

最后，通过基于 WCF 编写的网络视频管理系统软件进行设备探寻，该软件是针对管理 ONVIF 设备而编写的，由于现有管理平台的存在，希望集成后的系统能够直接与现有管理平台对接，从而实现管控功能。经过实验，该软件不能发现服务，因此也就不能使用该服务。经过调试与验证，共发现 2 个主要原因。

1) 该系统发出的“探寻”报文中，指定了设备类型为 NetworkVideoTransmitter，而服务元数据中指定的设备类型为 Device，两者类型不匹配。

2) 跟踪 2 种测试成功的软件，发现它们在收到“探寻匹配”报文中携带的服务端 UUID 后，再次调用“解析”函数解析出 UUID 对应真实的服务地址，从而获得元数据和服务引用端点。而该软件没有“解析”函数的逻辑存在，它通过 XAddrs 指定的地址进行直接引用。

因此需要完成如下 2 步修改。

1) 修改设备描述的元数据，将 Hosted 节点的 Types 字段设置为 NetworkVideoTransmitter。

2) 修改 WS4D-gSOAP 源码，在 ws-discovery.c

的 wsd\_gen\_ProbeMatches 函数中添加如下代码：

```
XAddrs=ws4d_targetep_get_XAddre(target);
If(Xaddrs) probeMatch->wsd__XAddrs*=(char*)
XAddrs
```

代码修改完毕后，通过 CMake 重新安装 WS4D-gSOAP，即可在 ProbeMatch 中拥有 XAddr 字段，使得视频管理系统软件能够发现服务。

至此，3 种软件都可以发现服务。经过简单改进，设备可同拥有业务处理能力的 ONVIF 标准软件具备互连、互通、互操作的能力。最后，可成功通过 DPWS Explorer 查看表示网络摄像机服务的信息元数据。

集成完毕后，考察 DPWS 对性能的影响。该影响不考虑设备的发现、事件和安全机制，因为非 DPWS 的嵌入式设备应用接口本身不具备这些特性。

利用 soapUI 对嵌入式 Web 服务进行性能测试，它直接利用 Web 服务的 WSDL 文件测试 Web 服务接口，因此包含操作层面和模式层面的完整 WSDL 覆盖分析。向工具中添加 Web 服务所发布的 wsdl 文档后，即可进入测试平台。测试案例设置 60s 内模拟 10 个线程同时连续发送请求，测试结果如表 3 所示。

表 3 负载测试

时段/ms	最小响应时间 /ms	最大响应时间 /ms	平均响应时间 /ms
200	2	219	8.13
100	2	545	25.93
50	2	742	35.94
10	4	1 079	59.44

实验证明，即使在高强度、多用户的并发请求的环境下，系统仍能做出毫秒级的响应。究其原因是在原有应用接口之上增加的 Web 服务接口，主要负责轻量级 SOAP 消息的分组封装、分组解析与数据解析的操作，而这种解析是基于 XML 文档的，因此拥有快速和低开销的特性，对性能的影响也较低。

当然，本次测试选用的业务较为简单，几乎不存在业务逻辑方面的延时。实际系统部署时应注意每个应用接口的响应时间应是不同的，响应时间与业务逻辑的复杂程度有关，应认为这种复杂业务逻辑导致的系统缓慢与 DPWS 协议栈的实施无关。

最后，分析系统的扩展性、顽健性及普适性。

1) 直连的 DPWS 是完全分布网络结构，它使

得系统的扩展性仅仅依赖于网络物理设施，如 Ethernet, WLAN，因此，扩展性较高。

2) 考察系统中某个设备失效的情况，可发现系统具有较高的顽健性。因为设备可在 DPWS 完全分布的模式使得“单点失效”对系统完全没有影响。

3) 通过 DPWS 协议栈可以看出 DPWS 主要针对应用的层次，其基本消息是 SOAP 分组封装。有关发现、通知和安全的特性仅仅是基于 SOAP 消息的应用。而 SOAP 作为一种基于 XML 的技术，是一种轻量级的协议，在嵌入式设备中实现较容易，主要包括 SOAP 的解析与封装，其本质是对结构化文本的操作。同时，设备联网的共性问题包括了设备的发现、通知和安全等共性需求，因此，将 DPWS 协议部署在嵌入式设备中具有很高的普适性。

综上，基于 DPWS 并采用直连策略构建的系统使得设备具有发现、事件与安全的特性。同时，实施 DPWS 后，在大量并发请求的情况下，设备仍有较高吞吐量。且其扩展性、顽健性和普适性较其他框架均具有较大优势。其基于万维网服务的协议栈也更加成熟，且易于开发、理解。此外，DPWS 还具备多种开源方案及完善的开发工具支持，使得设备集成的效率大大提升。

## 6 结束语

DPWS 是实现面向设备服务体系的一种基础框架，相比其他框架 DPWS 具有可大规模部署、提供内置安全、网络媒介独立和即插即用特性。其基于万维网服务的技术特征使得高层的万维网服务可以应用到低层的嵌入式系统中。这种基于万维网服务、利用 DPWS 的设备集成技术可广泛应用于有线/无线、微小型设备/大中型嵌入式设备、互联网环境/物联网环境等各种不同应用背景和不同设备的集成过程中。

## 参考文献：

- [1] World Wide Web Consortium (W3C) Web Services Architecture[S]. 2004.
- [2] Devices Profile for Web Services (DPWS) Version 1.1, OASIS Standard[S]. 2008.
- [3] Web services addressing (WS-addressing)[EB/OL]. <http://www.ibm.com/developerworks/library/ws-add>, 2004.
- [4] Web Services Dynamic Discovery Version 1.1[S]. OASIS Standard, 2009.
- [5] Web services eventing (WS-eventing)[EB/OL]. <http://ftpna2.bea.com/pub/downloads/WS-Eventing.pdf>, 2004.
- [6] BALLINGER K, CURBERA F, SCHLIMMER J, *et al.* Web Services Metadata Exchange (WS-MetadataExchange)[S]. 2004.
- [7] Web Services Security: SOAP Message Security 1.1[S]. OASIS Standard, 2006.
- [8] WS4D: Web services for devices[EB/OL]. <http://www.ws4d.org>, 2012.
- [9] The ITEA initiative[EB/OL]. <http://www.itea-office.org>, 2009.
- [10] SIRENA: service infrastructure for realtime embedded networked applications[EB/OL]. <http://www.sirena-itea.org>, 2006.

## 作者简介：



金路（1988-），男，江苏兴化人，南京邮电大学硕士生，主要研究方向为信息网络。



沈苏彬（1963-），男，江苏南京人，南京邮电大学研究员、博士生导师，主要研究方向为计算机网络、网络安全、信息网络等。