

基于 OPC UA 的跨平台设备互联互通关键技术研究

代瑞鑫

杭州企智互联科技有限公司，浙江杭州，311200；

摘要：在工业互联网和智能制造快速发展的背景下，设备层的数据集成和互联互通成为构建数字化工厂的基础性工程。OPC UA（统一架构的 OLE for Process Control）作为新一代工业通信协议，具备平台无关、安全可靠、语义建模等优势，逐渐成为异构设备之间实现统一通信的核心支撑技术。本文围绕 OPC UA 在多平台设备互联中的关键技术展开研究，从通信模型、信息建模、协议网关转换等方面系统探讨其技术原理与实现路径，并结合典型场景提出面向边缘计算、轻量部署与跨平台对接的优化策略，验证其在实际应用中的可行性与推广价值。研究表明，基于 OPC UA 的互联方案可有效打破工业现场设备间“信息孤岛”现象，推动制造系统向标准化、智能化方向演进。

关键词：OPC UA；跨平台通信；信息建模；协议网关；设备集成；工业互联

DOI：10.69979/3041-0673.25.10.045

1 OPC UA 协议体系结构与通信机制解析

1.1 OPC UA 的多层通信模型与平台适配机制

OPC UA 作为面向下一代工业通信的标准协议，它的体系结构设计呈现出高度的模块化特点，同时还具备平台中立性。其整体模型涵盖了传输层、服务层、会话层以及信息模型层这四个层级，借助对不同功能逻辑的封装操作，达成了从底层传输一直到上层语义的完整抽象化处理。在传输协议方面，OPC UA 能够支持多种多样的绑定方式，像是基于 TCP 的二进制协议、依托于 HTTP(S) 的 Web 服务，还有 MQTT 等诸多标准消息格式，以此来保证它可以适配不同的系统以及网络环境。它的核心机制表现为客户端—服务器模式与发布—订阅模式同时存在，这样既照顾到了传统轮询式数据访问的需求，又能满足工业场景下事件驱动所要求的高效通信需求。另外，OPC UA 还支持 C/C++、Java、Python、C# 等一系列的多种语言接口，并且能够在 Windows、Linux、ARM 架构等主流操作系统之下进行部署运行，从而为实现从工业控制器一直到云端服务的全链条数据通路筑牢了十分坚实的基础。

1.2 安全机制与会话控制策略

在工业数据处于高频交互且多层传输这样的背景之下，OPC UA 所具备的安全体系，实实在在是它能被广泛接纳的极为重要的根基所在。其安全机制是从通信加密、身份验证、消息签名，再到访问权限控制这四个方面来构建起来的，能够较为周全地顾及到数据传输过程当中可能潜藏着的各类安全威胁。在通信这个环节，所采用的是将对称加密与非对称加密相互结合起来的方

式，凭借着基于证书的那种双向身份认证机制，以此来切实确保客户端和服务端二者之间身份是具有可信性的。从消息层面来讲，是通过数字签名的方式来保障数据具备完整性以及防篡改的能力。而在访问权限这块儿，OPC UA 对细粒度的角色授权以及用户权限模型做出了明确的界定，能够依据用户的身份去配置特定节点的读写权限，还能够对不同的角色开展操作方面的限制以及审计追踪等相关工作。借助于对安全策略进行灵活配置的这一举措，系统便可以达成通信加密级别和资源消耗之间的那种最优的平衡状态，进而保障其在工业现场进行部署的时候，既能具备实用性，又能确保其安全性。

1.3 统一信息模型构建与设备语义描述机制

信息模型层在 OPC UA 的设计里极具创新性，它一方面对数据的结构做出了定义，另一方面还赋予了数据相应的语义，能够支持在多维度下开展设备建模工作，并且实现语义关联。OPC UA 秉持着对象建模的思路，依靠节点（Node）、引用（Reference）以及属性（Attribute）来构建起树状的信息空间。每一个设备对象当中，除了包含有实时的数据值之外，还能够嵌入像是方法、事件、状态机等这类行为属性，以此达成对设备运行逻辑的高度抽象处理以及封装操作。借助扩展型类型定义机制，开发者有能力构建出专属于特定行业或者特定厂商的自定义信息模型，而且在这个过程中依然可以依照统一的语义规则来完成对数据的解析工作。需要注意的是，OPC UA 已经和多个领域标准，比如 PLCopen、ISA-95、AutomationML 等，建立起了信息模型映射方面的关系，这就为跨设备、跨协议的协同作业打下了关于建模的基础，从而在很大程度上提升了系统集成的效率以及

互操作性。

2 跨平台设备对接关键技术研究

2.1 异构设备接入机制与 OPC UA 封装适配技术

在当下工业现场普遍存在着多协议以及多代际设备混合使用的环境里,要达成统一且高效的设备接入机制,这实实在在是一项系统性的工程。现场的设备,在功能方面、接口方面、通信频率方面,还有数据格式乃至厂商标准这些层面上,都呈现出高度异构的状态,所以直接去对接 OPC UA 的话,其难度是比较高的。如此一来,构建那种面向异构协议环境的 OPC UA 封装适配技术,就变成了实现跨平台互联互通极为关键的核心环节啦。对于那些依旧采用 RS-232、RS-485 等串口协议的传统 PLC、仪表或者控制模块而言,可以先通过串口转以太网模块,把它们的原始数据封装成网络能够识别的格式,接着再依靠数据采集卡或者工业协议网关来对这些数据展开协议解析的工作。这些原始数据经过转换层的转换,变成标准变量映射结构之后,再由 OPC UA 服务器按照节点的形式去进行建模以及发布等操作,这样一来,也就实现了非标设备的“标准化接入”呢。

对于那些具备了一定嵌入式能力的新型智能设备而言,能够借助嵌入 OPC UA 轻量通信栈或者 SDK 的方式,把其内部的变量径直映射成为 OPC UA 对象,如此便能够天然地与统一建模以及服务访问机制相互兼容。在进行适配操作的过程当中,其中最为关键的技术难题就在于信息语义的一致映射这件事上,具体来讲就是要思考怎样把本地设备的地址空间、变量的命名规则、更新的频率以及数据类型和 OPC UA 标准节点空间构建起一一对应的关系来。与此同时,为了能够满足工业级所提出的实时性方面的需求,系统还应当要确保在封装过程当中具备低延迟以及高并发的能力,尤其是在高频采集这样的场景之下,怎样去减少缓冲所产生的延时,怎样避免出现数据拥塞的情况以及怎样防止信息丢失等问题,已然成为了封装系统优化工作当中极为关键的要害所在。

2.2 跨平台 SDK 集成与轻量客户端构建策略

OPC UA 协议能够具有普适性以及跨平台的能力,这主要是依靠其通信栈在实现过程中所具备的灵活性,同时也离不开软件开发工具包(SDK)所达到的集成效率。在处于多平台、涉及多语言并且存在多种硬件的环境之下,要是想构建出 OPC UA 通信功能模块的话,那就得按照不同场景所提出的具体需求,去挑选最为契合的 SDK 实现架构。open62541 来举例,它作为一种主流的开源 C 语言 SDK,有着轻量的特点,其结构是模块化的,

而且不存在对第三方的依赖情况,所以它是特别适宜被部署在嵌入式控制器上面的,对于边缘计算设备以及现场智能终端来说,它也是非常合适的部署选择;而 Eclipse Milo(这是通过 Java 来实现的)以及 FreeOpcUa(这是通过 Python 来实现的)呢,它们则是更加适合被运用在上层信息管理平台或者集成开发系统当中,能够对高层次的业务逻辑控制起到支持作用,并且也能实现服务封装的功能。

在客户端的构建事宜上,传统桌面客户端得要有节点浏览、订阅通知、历史回溯以及断线重连这些功能,与此同时,还得支持对多设备展开集中监控,要能进行日志记录,并且做好用户权限管理方面的工作。着眼于未来工业互联网应用的发展走向,WebSocket 的 OPC UA over Web 技术还有 WebAssembly 加上浏览器端渲染方案正逐步兴起,它们有着跨终端、能跨浏览器且零安装这样的特性,可给移动端或者轻量终端设备打造出可视化的、响应较为快速的操作界面。再者说,客户端构建的时候,务必要着重去考虑网络出现抖动、现场突发一些情况、通信发生断链等这类异常场景下的处理办法,应当凭借内置本地缓存、断点重连以及热更新等策略,来提高客户端的稳定性以及业务的连续性,进而为其在工业现场能够长期且稳定地运行给予有力支撑。

2.3 OPC UA 与工业协议网关融合技术

工业协议网关在多协议异构设备当中扮演着极为重要的角色,其作用恰似‘翻译器’以及‘协调器’一般,它已然成为实现 OPC UA 能够在众多场景下广泛铺开并深度交融所必不可少的中间件组件。当下的情况,存在着大量的工业设备依旧在运用像 Modbus RTU、Modbus TCP、Profibus DP、Profinet、CANopen 这类并非 OPC 协议的协议,特别是在那些相对老旧的生产线以及过程控制系统之中,此类设备从其原生的设置来讲,是没办法对 OPC UA 通信予以支持的。所以说,借助网关设备来针对这些传统协议展开采集以及解析方面的相关工作,并且进一步把这些协议所涉及的变量映射成为 OPC UA 节点对象,这无疑是达成协议融合的一条极为关键的路径。

高效的网关得要同时支持双向通信才行,也就是说它不光能够读取传统协议设备的数据,而且还能把这些数据发布成 OPC UA 服务;与此同时,它也能够接收来自 OPC UA 客户端的控制指令,并且把这些指令转换为目标设备可以去执行的命令格式,如此一来便构建起了完整的读写闭环。在技术实现方面,网关需要具备可配置化映射模板,要有内置通道缓存机制,还得拥有多线程采集能力以及事件过滤策略,只有这样才能确保在海

量设备并发访问的情况下,系统依然可以保持低延迟、高吞吐这样的运行状态。再进一步来讲,现代协议网关正在逐步融合边缘计算能力,它能够支持本地规则引擎、数据预处理以及 AI 推理模型部署等一系列的功能,进而在现场侧实现初步的数据判断以及告警推送,这样就能减轻中心服务器的负担,还能提升整体的响应速度。往后,随着工业网络朝着 TSN 与以太网/IP 协议全面演进,协议网关可就不再仅仅是单一的数据转换器了,而是会成为集数据治理、安全隔离以及通信优化于一身的智能中枢设备,它和 OPC UA 标准的深度融合将会变成打造真正“即插即用、即连即通”的工业智能设备生态的重要基础。

3 基于 OPC UA 的智能互联架构与应用实践

3.1 典型应用场景与部署方案设计

OPC UA 所形成的设备互联解决方案,如今已在离散制造、过程控制、电力系统以及物流调度等诸多领域获得了广泛的应用。就离散制造领域而言,把机床、机器人、传送带、AGV 等设备全部归入到 OPC UA 地址空间当中,如此一来便能够达成产线级别的统一调度以及监控功能。在过程行业方面,像石油化工、水处理、电厂调度这些具体场景中,OPC UA 发挥着采集众多仪表数据并且和 DCS 系统进行对接的作用,进而实现分布式控制与中心级决策之间的协同配合。从部署方案比较推荐采用“边缘节点 + 中心平台”这样的分层架构模式,也就是现场设备借助边缘代理来采集数据,随后将其转换成 OPC UA 格式,再上传到 MES/SCADA 平台上,以实现统一的可视化呈现以及管控操作。这样的结构不但能够确保数据具有实时性,而且还可以提升整个系统的灵活性以及容错性。

3.2 面向边缘计算的 OPC UA 轻量部署机制

边缘计算架构逐渐兴起,这一情况使得 OPC UA 面临着更高的适配方面的要求,需要在那些资源存在一定限制的嵌入式系统当中,去实现协议栈的正常运行以及数据的有效管理。有鉴于此需求的存在,轻量级的 OPC UA 代理组件便得到了极为广泛的开发,其具备内存占用量小、响应延迟低以及连接可靠性高这样的一些特性。该代理组件能够在 ARM 架构的控制器、Linux 工控主板以及智能 IO 模块等上面运行,可实时对设备的状态进行采集,并且以发布/订阅的模式朝着上游系统去推送相关的事件。与此同时,边缘节点自身也能够在本地完成诸如报警判断、数据缓存以及模型推理之类的处理任务,如此一来,便能够在很大程度上减轻中心服务器所承担负担。要是结合 Docker 等相关的容器技术的话,

那么就可以实现 OPC UA 组件在边缘侧的快速部署操作、热更新操作以及动态扩缩容操作,进而为智能化的运维工作提供更为高效的基础性支撑。

3.3 推进标准协同与生态融合的发展路径

要推动 OPC UA 在设备互联互通方面得到广泛应用,这可不单单取决于技术层面的优化工作,还得让生态和标准协同起来共同发展才行。当下,像 OPC 基金会、工业互联网联盟(IIC)以及德国工业 4.0 平台等国际标准化组织、行业联盟,都已经把 OPC UA 列入了核心推荐协议当中,并且围绕着它去构建起适用于多个行业、多种场景的信息模型规范。在国内呢,也应当进一步强化 OPC UA 和国产工控设备、国产 PLC 以及工业软件系统之间的对接适配程度,以此推动那种“标准模型加上本土化实现”的协同机制能够有效运转起来。与此同时,还应当去鼓励形成可以开放共享的行业模型库以及信息建模工具平台,把中小企业使用的门槛给降低下来,从而构建起一个以 OPC UA 作为桥梁的数字化制造开放生态环境,推动我国的工业系统朝着平台化、服务化以及智能化的方向实现全面的转型升级。

4 结语

伴随工业数字化以及智能化不断向前发展,让设备层达成互联互通的状态,这已然变成制造系统朝着智能协同方向演化极为关键的基础所在。OPC UA 属于面向下一代工业通信的核心协议,它凭借自身具备的平台无关这一特性、有着服务导向的结构、安全机制十分完备以及信息建模的能力颇为强劲等诸多技术方面的优势,正在一步步地破除不同厂商、各类协议之间所存在的壁垒,进而成为构建那种开放、符合标准且高度集成的工业互联体系相当重要的支撑力量。从通信模型一直到信息语义层面,从协议适配再到边缘部署环节,OPC UA 在实际的工程场景当中都彰显出了很强的技术适应性还有落地实施的能力。通过引入能够跨平台的 SDK,构建起可以融合多协议的网关,推动实现轻量化的部署并且促进行业模型标准协同起来,OPC UA 在达成数据的互通、实现设备的集成以及促使系统协同这些方面,均展现出了极为广阔的应用前景。

参考文献

- [1] 禹鑫焱,唐权瑞,施甜峰,等. 基于 OPC UA 协议的工业网关系统设计与实现[J]. 高技术通讯. 2021, (9).
- [2] 王洲,牛鹏飞,王钧. 基于 OPC UA 的制造资源虚拟映射模型实现技术研究[J]. 中国仪器仪表. 2021, (10).
- [3] 杨军平,孙志敏,赵巧玲. OPC UA 技术在工业物联网中的应用及趋势[J]. 新技术新工艺. 2020, (11).