

关注边缘计算产业联盟  
请扫二维码

# 支撑边缘智能计算的软件体系： 语言、工具与信息框架

Enabling Intelligence in Edge Computing From Software Perspective: Language, Tools and Informatics Framework

★上海交通大学 戴文斌

**摘要：**边缘计算是智能制造的基石，灵活的软件体系是支撑边缘智能计算的核心。边缘计算同时也是传统工业由自动化向信息化与智能化发展的重要手段。在边缘计算的推动下，传统工业现场设备的软件系统势必发生翻天覆地的变化。本文通过对现有边缘端工业软件体系的剖析，分析边缘计算给工业软件设备层带来的需求变革以及未来发展趋势。

**关键词：**服务化边缘计算；建模语言；设计方法；动态重构；工业软件智能体；边缘机器学习；数据采集监控

## 1 引言

上世纪70年代，可编程逻辑控制器的诞生促使工业进入3.0时代，制造过程自动化程度得到大幅度的提升。过去十年中，随着计算机与信息技术的高速发展，工业数字化得以迅速普及。工业软件是企业生产过程实现自动化与信息化的关键，涉及到设计、编程、工艺、控制、监控、通讯、管理等所有环节。工业软件往往需要根据过程控制、运动控制、离散制造等不同行业的工艺需求进行定制，以提升企业生产效率、优化配置资源以及提升产品的质量。

现有的工业软件大体上可以分为嵌入式软件与非嵌入式软件两类。嵌入式工业软件主要是应用在可编程逻辑控制器、传感器、工业网关、机器人、AGV等

工业现场设备中，实现对生产过程的状态采集、实时控制与通信，通常对实时性、安全性、可靠性有着极高的要求。而非嵌入式软件则为包括产品的研发设计与业务的运行管理提供信息化手段，这类的工业软件主要包括计算机辅助设计软件CAD、产品数据管理系统PDM、产品生命周期管理系统PLM、企业资源计划系统ERP、客户管理系统CRM、生产执行管理系统MES、数据采集与分析系统SCADA、系统仿真软件SIM等等。

现有的工业系统多数遵循ISA-95国际标准，如图1所示，ISA-95为工业系统定义了类似于金字塔结构的五层框架。金字塔五层框架的最底层是现场层，现场层通常包含各类传感器、执行器、变频器、电机等现场设备。在设备层之上是现场控制层，工业自动化系统通常采用可编程逻辑控制器（PLC）或者分布式控制系统（DCS）作为控制大脑。PLC以及DCS通过工业现场总线与现场层的传感器与执行器连接形成传感器数据→控制逻辑→执行器指令的闭环结构。PLC主要应用于机器设备的实时控制、运动控制、过程控制等领域，而DCS则主要应用在复杂过程控制中。IEC 61131-3国际标准定义了五种PLC的编程语言，分别包括图形化编程语言Ladder Diagram（LD）、Sequential Function Chart（SFC）和Function Block Diagram（FBD），以及文本型编程语言Structure Text（ST）和Instruction List（IL）。连接PLC与DCS的是监控层，通常监控数

据采集系统（SCADA）会部署在每个工厂，允许用户从PLC与DCS采集过程数据，提供历史数据趋势、实时与历史警报、图表报告、人机交互界面等功能，并且支持下载参数到PLC与DCS上。在监控层之上的是管理层，生产执行管理系统（MES）负责工厂制造数据管理、生产计划排程、生产调度管理、仓库管理、质量管理等功能。金字塔的顶层是企业资源管理系统（ERP）系统，通过对企业全方位信息的汇总与优化，提升企业管理效率，降低成本，有较大的经济价值。

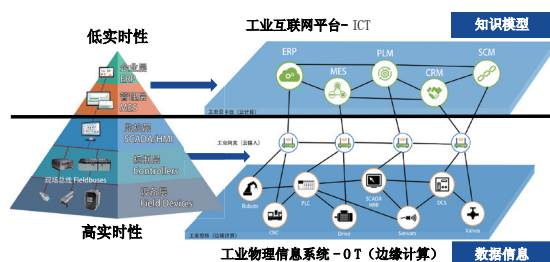


图1 工业系统框架

根据ISA-95标准制定的金字塔结构同样存在许多弊端。首先其中的不同层级间的信息传递十分困难。例如，当ERP系统需要读取一个工厂内传感器的数据时，需要经过一个复杂的数据交互过程，首先ERP系统向MES系统发送请求，MES转而向现场SCADA系统请求数据，SCADA系统则需要从相应的PLC或者DCS系统查询变量，而PLC或者DCS系统则负责从传感器获取最新数据返回给SCADA系统，进而传递给MES系统，最后才能抵达ERP系统。试想一下在大型系统中当数千个参数以这种方式传递时，整个信息的传递过程不但耗时且效率非常低下，造成大量计算与通讯资源的浪费。

其次，固化的软件结构无法适应工业互联网时代下大规模定制生产的需求。在工业互联网平台的支撑下，客户可以根据原料、工艺、功能等多方面的因素自定义产品需求。边缘端需要根据不同的订单要求动态地配置组合本地资源以完成生产任务，而现有的构架需要对软件进行长时间停机调试，造成停工时间大幅度增加。

除此之外，边缘端设备与设备间缺乏有效协作手段，无法应对不断变化的生产状态。ISA-95框架仅仅提供了纵向集成手段，而对横向集成，尤其是设备间与系统间的协作并未涉及。现实情况是每家厂商拥有自己的标准与技术手段，使得不同边缘端平台间协作十分困难，缺乏统一的有效手段来解决兼容性问题。因此，工业互联网对边缘计算软件提出了全新的需求。

因此，边缘端的软件智能化就变得尤其重要。如果说边缘计算是智能制造的基石，那灵活的软件体系就是支撑边缘智能计算的核心。边缘计算同时也是传统工业由自动化向信息化与智能化发展的重要手段。在边缘计算的推动下，传统工业现场设备的软件系统势必发生翻天覆地的变化。

## 2 边缘计算下的软件构架需求

软件定义一切（Software-Defined Everything，SDX）的概念已经深入企业，正如许多人所认同的，将来所有的工业企业都是软件企业，也都是数据企业。软件定义的核心特征是灵活性（Flexibility），灵活性可以体现为软件的功能行为能够根据需求进行动态调整而无需重新编程。软件定义的边缘计算赋予工业互联网对未来无限可能性（Future-Proof）的拓展可能，为现有的工业转型带来活力。

灵活的体系也对边缘计算软件提出了许多新的需求，如图2所示。首先，边缘计算下的工业软件必须具备动态重构（Dynamic Reconfigurability）的特征。动态重构能帮助系统针对需求变更做出实时自我调整生产过程。动态重构的成熟度可以分为多个等级：低级的动态重构可以根据需求变更调整工艺参数，受限于固化的生产过程，低级的动态重构仅能满足一部分简单的定制要求，例如调整过程控制中各种参数；中级的动态重构在调整工艺参数的基础上，应具备从预先设定的功能集合中动态组合更新生产过程从而满足定制需求，比如在离散制造过程中选择所需的工位，虽然通过选择功能可以一定程度上实现产品定制，但仍然缺乏学习新的生



关注边缘计算产业联盟  
请扫二维码

产工艺流程的能力,无法做到真正的“随心所欲”;高级的动态重构则不但能完全按照客户需求对软件动态组态,而且能创造新的功能与定义新的工艺流程。在无需人工干预的情况下,产线上的每个产品都能做到定制而不用人工干预调整软件,才是真正的智能制造。

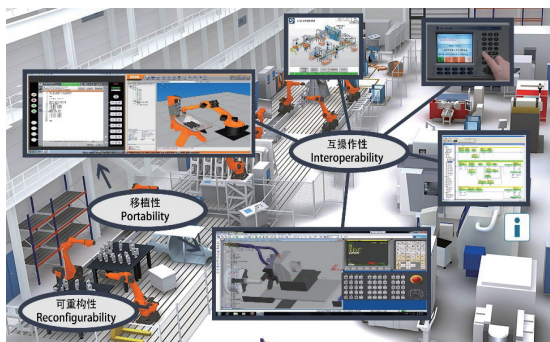


图2 边缘计算下的软件构架需求

边缘计算下的工业软件必须实现可移植 (Portability), 具备从一个平台向另一个平台迁移的能力。工业软件兼容性是多年来一直困扰着业界的难题,以工控软件为例,虽然绝大数厂商声称支持 IEC61131-3 标准,但是这些厂商的软件产品相互之间并不兼容,假设如果一家企业基于西门子 Step7 开发了大量的应用库,这些代码很难被移植到罗克韦尔或者施耐德的平台上。各家厂商一方面为了保护自己的知识产权与客户资源设置壁垒,另一方面是对标准的理解并不相同使语言的解析各不相同,造成看起来完全一样的代码在不同的平台上运行也可能产生多种结果,从而造成软件执行结果的不确定。

边缘计算下的工业软件必须能够互联互通相互操作 (Interoperability)。现有的工业系统通常使用工业现场总线来进行通讯,而每家厂商都有研发了专属的总线协议与标准。目前,常用的工业现场总线协议就有四十余种,如此多的总线协议造成设备间无法正常进行信息交互。即使设备与设备间使用相同的现场总线,也存在信息模型交互困难。OPC UA 的出现为实现设备间的互操作提供了通用信息模型,且定义了访问方式。在

此基础上,边缘端仍需要进一步结合知识经验进行实时决策,从而实现真正的智能制造。

在工业互联网与边缘计算的共同推动下,软件定义的工业软件须向新的体系构架进化来满足大规模定制的需求。

### 3 面向服务的边缘计算软件框架

服务化的软件系统在计算机领域已经得到了广泛的应用。相对面向对象 (Object-Oriented Programming) 等其他编程范式,面向服务的构架 (Service-Oriented Architecture, SOA) 有着松散耦合的特性,极大提升了系统的灵活性。SOA 主要分为三个部分:分别是服务提供方、服务需求方以及服务源 (Service Repository)。服务提供方将自己所提供的服务内容、访问方式等内容写入服务合约 (Service Contract) 中,并将服务合约发布到服务源。服务需求方当需要某项服务时,会先向服务源发送查询请求,当服务源匹配到合适的服务提供方时,会将此服务合约发送给服务需求方。需求方根据服务合约内的所约定的访问方式动态创建与服务提供方的连接。SOA 最大的特点就是迟绑定 (Late Binding),即当需要请求软件服务时,供需双方才建立实时连接,当执行完成时,绑定会立即解除。软件框架如图3所示。

面向服务构架的松散耦合特性能满足边缘计算下软件实时变更的需求,然而有些最基本的问题必须要解决,例如:

- 服务提供方如何让需求方了解所提供的服务;
- 服务需求方如何发现他们所需要的服务;
- 服务提供方如何授权给需求方服务的执行权限;
- 服务需求方如何通过服务编排来组合调用多个服务。

边缘计算的软件服务必须包含两个部分:核心服务以及扩展服务。核心服务回答了以上问题:

- 发现其他服务的能力;
- 实现服务提供方与需求方之间松散耦合的数

据交换;

- 授权服务使用;
- 自动编排服务功能。

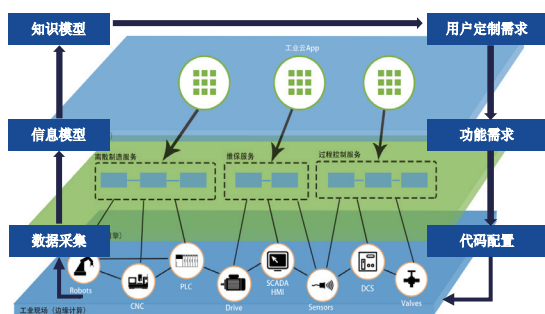


图3 面向服务的边缘计算软件框架

而扩展服务完全取决于服务提供方,只需符合服务标准接口协议即可。在边缘计算中,每个节点都可以作为服务的提供方,同时也是服务的需求方。每个节点根据自身硬件条件限制将不同服务分门别类,例如传感器可以提供各种温湿度、振动、到位等信息,控制器则可以根据各种传感信息提供反馈控制信号,而执行器(变频器、电机等)则根据指令驱动硬件。控制服务则同时需要传感服务提供的数据以及调用执行服务来实现实时控制。在边缘计算系统中,存在着多个基础型服务提供节点间基本信息交互,同时也有组合型服务,即通过对基础型服务的编排实现复杂功能,例如嵌入式机器学习服务可以根据传感器数据来对产品质量进行实时分类。复杂的服务编排在非功能性需求中有着广泛的应用前景,例如系统安全、实时性或者可靠性需求等。边缘计算软件服务化的实现路径有很多,例如COAP、XMPP或者OPC-UA协议都可以满足服务化的需求。

在复合型服务之上,更多的软件服务通过编排共同完成工艺流程,我们称为应用(Application)。这几年工业互联网与工业App的概念随处可见,工业App的本质是工业软件的服务化,与传统手机App终端与云端互动的构架所不同的是,工业App的形式不仅仅是简单的一对一连接,也可以由多个不同设备与云端协作完成。而边缘计算则是工业App的落脚点,

仅靠云端的工业App是瘸了腿的兔子,无法对生产系统做出实时优化,同样也无法产生更大的经济价值。因此边缘计算应用可以小到一个工位的控制与人机界面操作,也可以大到整条产线的控制系统。边缘计算应用之间通过协作也可以形成更大的子系统(System of Systems),使用统一构架的边缘计算子系统相对现有的ISA-95构架拥有更强的延展性、灵活性与较强的兼容性。

## 4 边缘计算通用建模语言

实现服务化边缘计算系统的一大障碍是多种编程语言的共存。现有工业系统内各种嵌入式软件五花八门,例如C/C++语言、IEC 61131-3所包含的五种PLC编程语言,各种.Net/HTML5/JavaScript等人机界面编程语言,甚至于使用python作为嵌入式机器学习等等。除此之外,设备与设备之间的信息通讯也不畅,单控制器与传感器之间所使用的工业互联网总线就有二十余种。最后,可视化的模型同样也是必不可少的。可视化建模语言能够提供给用户直观的系统设计,特别是对高复杂性的系统,抽象化模型可以提升系统设计的效率。因此,目前边缘计算缺乏能够涵盖感知、控制、通讯、监控、数据处理等多种用途的通用可视化建模语言,如图4所示。

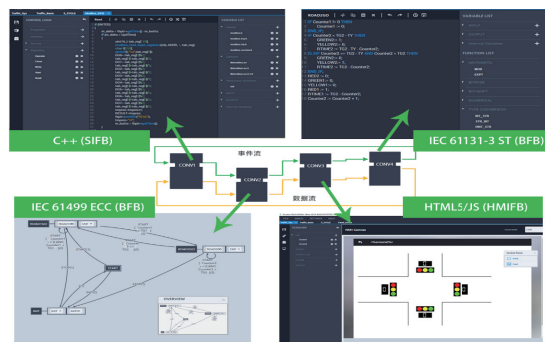


图4 边缘计算通用建模语言

如上一章节所叙述的,边缘计算服务框架能够赋予边缘计算系统可移植、可重构与互操作等特



关注边缘计算产业联盟  
请扫二维码

性。结合边缘计算编程语言的实际情况,模块化的设计(Component-Based Design)是实现服务化框架的最佳选择。模块化设计的基本元素是功能块(Function Block)本体与通用软件接口(Interface)。使用基于IEC 61499功能块标准封装IEC 61131-3 ST、LD、C/C++、HTML5/JS等编程语言,并且将功能块的输入与输出接口分别映射成为软件服务的输入参数与返回参数,是实现边缘计算软件服务化的有效手段。

IEC 61499作为可执行的建模语言提供了标准的功能块接口定义、分等级的功能块网络、部署模型以及管理协议为模块化抽象系统设计提供了强有力的支持。基础型服务可以使用IEC 61499基础功能块表述,每个基础功能块可以定义多个逻辑(Algorithm),IEC 61499标准并未指定编程语言,因此每个逻辑可以使用不同的编程语言编写。在此基础上,每个逻辑可以单独作为一个服务入口罗列到服务合约中,当此软件服务被调用时,服务需求方将更新输入变量,而服务提供方则将执行相应的逻辑代码并且返回结果(输出变量)。IEC 61499标准同时也提供了复合功能块类型,每个复合功能块则内含功能块网络,结构化分层级的服务编排有了用武之地。IEC 61499标准的部署模型允许在同一个系统内设置多个并行的应用,并且多个应用可以分别运行在不同的设备上,而每个应用中的功能块网络也可以分别部署到不同的设备上。此部署模型将复杂的设备间数据交互与部署工作抽象化,并且可以通过管理协议动态部署且直接执行,减少了人工配置造成的错误,大幅度提升了系统开发的效率。

## 5 边缘计算数据驱动的智能管控

得益于芯片技术的发展,边缘节点的计算与储存能力得到大大增强,在满足实时控制的同时,剩余的算力与储存能力还没得到充分挖掘。随着工业互联网的推广,数据采集与分析的需求成爆炸式增长。毫秒级的传感器数据采集频率能够产生大量的生产过程数据,若将数据完全搬到云端会造成极大的计算与储存压力,造成企业云计算成本直线上升。在边缘计算对数据进行验

证、清洗、过滤、缓存能够大幅度缓解云端的压力。此外,高实时性的警报、机器学习模型判断、甚至于机器学习模型训练等任务可以通过云端分发到边缘端,在数据源头直接进行判断,基于工业多软件智能体、知识推理等以往需要大量计算与储存资源的技术满足高实时性系统决策的需求,在边缘计算系统中也有大量发挥的余地。边缘-云协作模式下,能突破以往系统的边界,激发更多新的业务模式创新与工艺水平提升。

## 6 结语

相对于云计算的低实时性与高性能计算的特点,边缘计算善于处理高实时性小数据。结合边缘计算与云计算的特点,工业互联网应采取云计算与边缘计算相辅相成的闭环策略。相对于云计算的成熟体系,边缘计算建立完整的语言、工具与标准体系刻不容缓也势在必行。**AP**

### 作者简介:

**戴文斌**,上海交通大学电子信息与电气工程学院自动化系副教授,教育部系统控制与信息处理重点实验室成员。分别于2007年与2012年在新西兰奥克兰大学获得荣誉学士与博士学位。近年来主要从事基于分布式工业自动化系统软件架构体系设计与应用、工业信息物理系统信息化与知识自动化、医疗物理信息融合系统与知识推理等方向的研究工作。以第一作者身份在IEEE Transactions和国际重要学术会议上发表SCI/EI论文40余篇,合作出版英文专著1部。作为子课题负责人,参与完成了欧盟最大物联网与自动化项目Arrowhead;参与个性化硬组织精准医疗器械的智能设计及增材制造体系构建、智能电网运行及交易理论、工业控制系统自动软件构造等科技部国家重点研发计划;获得2013和2014 IEEE IES年会最佳报告奖。目前,为IEEE高级会员,IEC SC65B WG15标准以及IEEE P2660.1标准制定工作小组成员,担任IEEE IES工业智能体、工业信息学技术委员会成员,IEEE-IES上海分会秘书,IEEE SMC智能工业系统技术委员会成员。