高技术通讯 2018 年 第 28 卷 第 6 期: 553 ~ 559

doi: 10.3772/j.issn.1002-0470.2018.06.009

基于消息代理的 OPC UA 发布/订阅模式研究与实现①

刘 洋②*** 刘明哲* 徐皑冬* 王 锴* 韩晓佳*** 张 凝*** 滕玉坤*** 江俊慧***

(* 中国科学院沈阳自动化研究所 沈阳 110016)

(** 中国科学院大学 北京 100049)

(*** Department of Electronic Systems Engineering, Hanyang University, Ansan 15588 Korea)

摘 要 针对目前 OPC UA 客户端/服务器通信模式中存在的紧耦合、服务器性能瓶颈等问题,进行了 OPC UA 发布/订阅通信模式总体架构的研究。首先具体分析了 UA 的核心功能,包括地址空间技术和数据编码技术,并在两者的基础上,利用消息代理机制,完成了发布者和订阅者的功能开发。还通过实验进一步验证了功能的稳定性和对数据的传输能力,结果表明,此模式可以满足大多数工业需求。

关键词 OPC 统一架构, 地址空间, 数据编码, 发布/订阅, 消息代理

0 引 言

在工业控制领域,存在着各式各样的自动化系统,这些系统又大多来源于不同的生产厂家,遵循着不同的通信协议,这使得各个系统之间难以进行信息交换,形成大量的“信息孤岛”。OPC 统一架构(OPC unified architecture,OPC UA)是解决上述问题的有效方法之一[1]。OPC UA 技术支持复杂的数据类型,具有统一的地址空间、跨平台传输以及特有的安全机制等特点。利用 OPC UA 技术,不仅能够实现系统中的同层设备之间的信息交换,同时也能够实现从工业现场设备到高级控制层的纵向信息集成,在各层信息交换的基础上,进一步实现工业系统的互操作性[2,3]。

目前,OPC UA 技术仍然采用客户端/服务器(client/server,C/S)通信模式[4,5],此种模式中,UA 客户端直接连接到 UA 服务器进行请求与响应过程,造成了客户端与服务器应用程序之间的紧耦合,而且 UA 服务器存在资源限制,不允许连接过多的客户端。

针对 C/S 模式中存在的问题,OPC 基金会在 2017 年最新颁布了 OPC UA 发布/订阅模式[6]。与 C/S 模式不同,发布/订阅模式中的 UA 应用程序不需要直接交换请求与响应,而是通过消息中间件完成。OPC UA 发布/订阅模式可以实现客户端与服务器的解耦,同时具有良好的可扩展性,可实现多个客户端与多个服务器的连接。并且随着云计算时代的到来,以手机、平板电脑介质为代表的移动终端应用最终会被广泛使用,OPC UA 的发布/订阅模式也更适合移动应用等的远距离传输需求。

本文设计了一种基于高级消息队列(advanced message queuing protocol,AMQP)的消息代理系统模型来实现 OPC UA 发布/订阅通信模式。

1 OPC UA 简介

OPC UA 规范共由 14 个部分组成,第 1 至 7 和第 14 部分详细说明了 UA 的核心功能,包括地址空间、数据编码、第 4 部分由服务定义的 C/S 模式以

① 国家自然科学基金国际(地区)合作与交流(71661147005)资助项目。

② 女,1994 年生,硕士生;研究方向:工业通信技术、嵌入式系统等;联系人,E-mail: liuyang3@sia.cn

(收稿日期:2017-12-29)

及第 14 部分定义的发布/订阅模式。第 8 到 11 部分将这些核心功能应用于特定类型的访问，定义了存取类型规范。第 12 部分描述了 UA 发现机制以及第 13 部分描述了聚合数据的方法。本文详细介绍 UA 的核心功能。

1.1 地址空间

地址空间是 OPC UA 的核心概念[7]，UA 的其他功能都需要在地址空间的基础上实现。地址空间的基本组成单位是节点，节点是实际设备在地址空间中的映射，地址空间为不同的实际设备提供了一个统一的抽象模型，利用这个抽象的模型，便于实现对节点的统一管理[8]。

节点由属性和引用两部分组成，图 1 显示了节点的结构和节点之间的关系[7]。节点可以根据不同的用途归属于不同的节点类别，属性被用来描述节点的一些特性，根据节点类别，一个节点可以有不同的属性集。但是，每个节点也有一些共同的通用属性，表 1 对这些属性进行了总结。

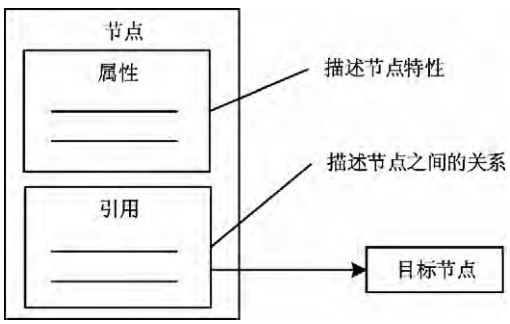

图 1 地址空间的节点模型

表 1 节点的通用属性

属性	描述
NodeId	唯一确定地址空间的节点
NodeClass	定义节点所属类别
BrowseName	浏览地址空间时用于标识节点
DisplayName	用于向用户显示的节点名称
Description	节点的文本描述
WriteMask	规定是否可修改节点属性
UserWriteMask	当前用户是否可修改节点属性

引用描述了两个确定节点之间的关系，并将节点组织成层次结构。引用可以理解为：存在于一个节点中的指针，通过保存其他节点的 NodeId 来指向另一个节点。通过节点间的引用可以把地址空间组织成网状结构。但是为了提高互操作性，一般建议将地址空间的节点以层次结构组织，即类似于 Windows 的文件夹结构。本文介绍 3 种最重要也是最常用的引用类型，分别是 HasComponent、Organizes 和 HasSubtype。

HasComponent 用于定义对象所包含的组件，表示包含的关系，目标节点是源节点的一部分，目标节点必须依赖于源节点而存在，当源节点被添加到地址空间后，它所包含的组件也会随之一起添加到地址空间里；Organizes 用于定义树目录的层次结构，组织地址空间中的节点，提供浏览地址空间的便利机制；HasSubtype 用于定义类型继承，表示类型的子类型关系，用来跨越引用类型层次结构。

1.2 数据编码

数据编码是将服务消息包括它自身的输入、输出参数序列化到网络格式，目前，OPC UA 规定了两种编码方式[9]，分别为 UA 二进制编码和 XML 编码。

在工业控制系统中，实时性和线路开销往往是人们关注的重点问题，随着设备集成化的快速发展，数据传输量日益增加，因此 UA 工作组定义了 UA 二进制编码以适应工业现场环境[10]。UA 二进制编码可以实现在数据线路上占用较小的体积，并且提供快速的数据编码和解码，本文重点关注 UA 二进制编码方式。

OPC UA 二进制的基本概念是基于一个明确定义的规则，将特定的内置基本数据类型翻译成二进制表示形式，并按照一定的顺序写入一个二进制流。UA 二进制编码包括编码过程和解码过程，编码过程是将数据信息按照规则编写成为二进制流的形式，解码过程是编码过程的反过程，将二进制流重新转换为数据信息的过程，如图 2 所示。

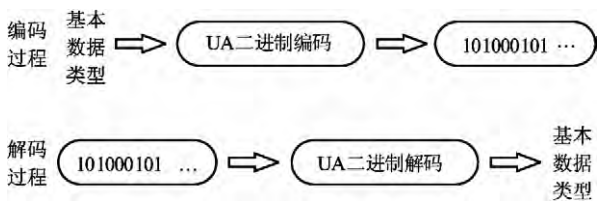

图 2 UA 二进制编码解码原理图

1.3　客户端/服务器模式

由服务定义的客户端/服务器(client/server,C/S) 通信模式主要表现为请求响应过程,如图 3 所示,客户端向服务器提交请求消息,在服务器收到请求后,分两步处理消息: (1) 解码消息,并找到要执行的服务; (2) 如果成功解码消息,则执行该请求服务操作,并生成一个成功或失败的数据,包含在要返回的响应消息中。

图 3　客户端/服务器架构

1.4　发布/订阅模式

在 OPC UA 的发布/订阅模式中,UA 应用程序之间不会直接交换请求和响应,而是发布者将消息发送到消息中间件,订阅者通过消息中间件去订阅它所感兴趣的消息,如图 4 所示。系统内的各个发布者和各个订阅者之间的通信完全通过消息中间件完成,各个发布者和订阅者之间是透明的,它们各自并不知道对方的存在。

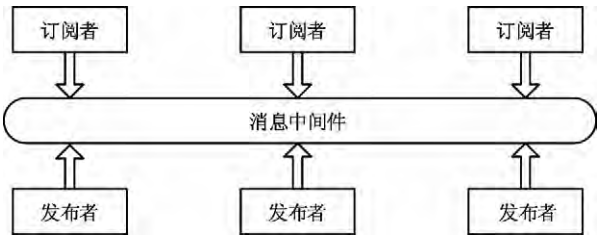

图 4　发布/订阅架构

为了适用于各种各样的应用案例,OPC UA 发布/订阅模式中的消息中间件支持两种机制,分别为无消息代理机制(broker-less form) 和基于消息代理机制(broker-based form) ,本文重点研究与实现基于消息代理的机制。

基于消息代理的机制的发布/订阅模式可以最大程度地实现 UA 应用程序的解耦。发布者发送消息时,订阅者可以处于连接断开状态,此时如果有发布者发布了此订阅者订阅的消息,消息代理会负责存储消息,直到订阅者重新连接上消息代理。由于它们之间的这种透明性,发布/订阅模式可以动态改变发布者和订阅者的数量,这种特性使得 OPC UA 可以更好地适应工业分布式系统。

2　消息代理机制模型

高级消息队列协议(advanced message queuing protocol,AMQP) 是一个标准开放的消息队列协议,基于此协议的客户端与服务器可以互相传递消息,AMQP 的设计目标是允许来自不同生产厂家的消息生产者和消费者实现真正的消息共享。

AMQP 模型[11] 如图 5 所示,发布端负责消息的产生,订阅端负责从消息队列中获取消息,AMQP 服务器中包括两个主要的组件: 交换器和消息队列。交换器接收发布端发送的消息,并根据设定的路由规则将消息分发到消息队列,消息队列存储消息,直到这些消息被订阅端接收完毕为止。绑定规定了交换器和消息队列的关联,提供消息路由的规则。

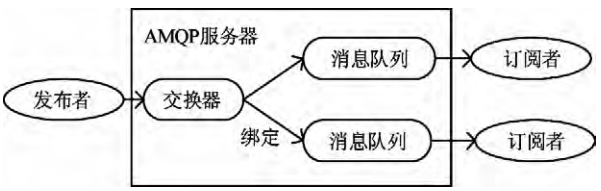

图 5　AMQP 模型

AMQP 定义了 4 种类型的交换器,分别为 Fanout、Direct、Topic 和 Headers 类型。

(1) Fanout 交换器: 广播式交换器,无论消息的路由关键字是什么,这条消息都会被投递到所有与该交换器绑定的队列中;

(2) Direct 交换器: 直接式交换器,精确匹配消息的路由关键字和绑定关键字,将消息投递到一个或多个队列中;

(3) Topic 交换器: 主题式交换器,模糊匹配消息的路由关键字与绑定关键字,将消息投递到被绑定的队列中。Topic 交换器常被用于发布/订阅模式中[12] 。

(4) Headers 交换器: 交换器与消息队列之间的绑定通过一系列“属性 = 值”的键值对来表示,消息的路由值以“属性 = 值”集合的方式定义,被发送到

与之相匹配的队列中去。

3 OPC UA 发布/订阅模式架构

3.1 系统总体设计与实现

本文设计了基于 AMQP 的 OPC UA 发布/订阅模式总体架构，如图 6 所示，本架构中具体包括了 4 个模块：地址空间模块、发布者模块、AMQP 服务器模块和订阅者模块。

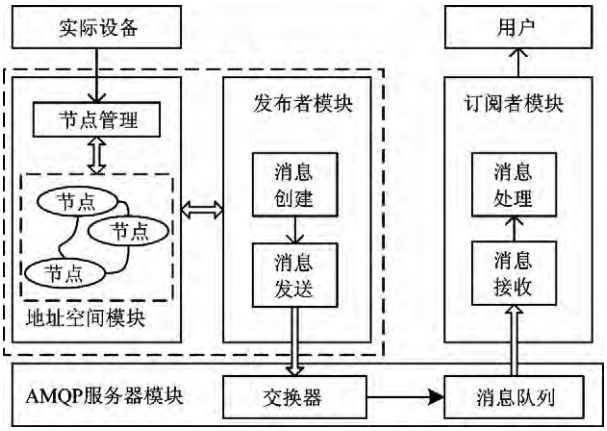

图 6 发布/订阅模式总体架构图

3.1.1 地址空间模块

地址空间中的节点包含了实际设备的具体信息，可通过节点管理对节点进行修改、增加、删除等工作。图 7 显示的是实际编程时节点的组织情况，在根节点下有对象、类型和类节点视域。对象下面组织的就是地址空间的主体，表示了实际设备在地址空间的组织，图中示例中给出了两个控制器实例，分别为 controller1 和 controller2。类型包括了地址空间中的类型，每个节点都引用其中的一个类型节点，图中示例给出了控制器类型 controllerType。控制器实例便是控制器类型的子类型。

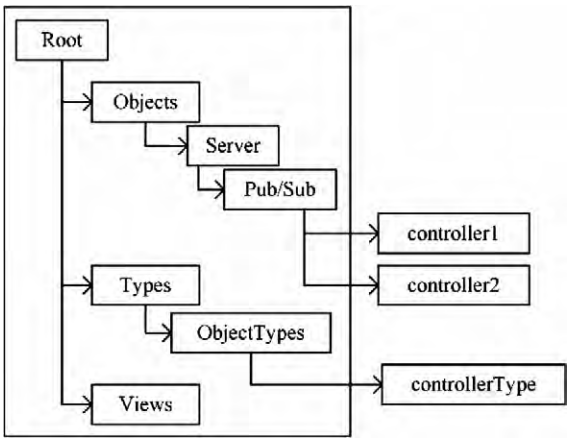

图 7 地址空间实际架构

3.1.2 发布者模块

发布者模块首先需要完成消息创建的功能，具体创建流程如图 8 所示。

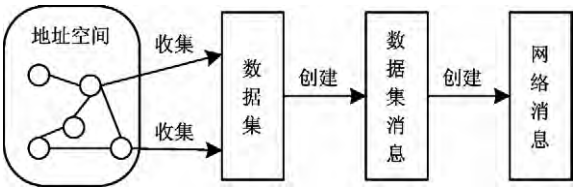

图 8 网络消息创建流程

第一步是收集要发布的数据，并在收集过程中配置发布数据集参数，如数据集序号(DataSetClassID)和数据集元数据等字段信息，生成数据集。

第二步是创建数据集消息，接收上一步产生的数据集，并进行 UA 二进制编码，创建生成数据集消息。UA 二进制编码的特点如下：OPC UA 一共支持 25 种内置数据类型，本文仅以其中常用的两种数据类型，整型和字符串型举例说明实现 UA 二进制编码方式。

整型：OPC UA 中的所有整数类型，包括有符号，无符号，32 位和 64 位整型都被编码为小端格式，即最低有效字节出现在字节流的前端。图 9 解释了值 2000000000(Hex：77359400)的编码。

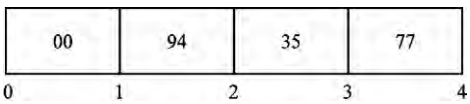

图 9 整型编码

字符串型：OPC UA 中的所有字符串值被编码为 UTF-8 字符的一个序列，并以字节流的长度开始。图 10 解释了字符串“OPC UA”的编码。

5				O	P	C	U	A
05	00	00	00	4F	50	43	55	41

图 10 字符串编码

第三步是创建网络消息，接收上一步的数据集消息，配置发布者序号(PublisherID) 等，生成网络消息。

发布者还需完成消息发送的功能，需将产生的网络消息发送给 AMQP 服务器，发送的工作流程如下。

(1) 创建一个频道(channel) ，连接到 AMQP 服务器端;

(2) 创建一个交换器，设置交换器类型以及相关属性;

(3) 创建一个消息队列，设置相关属性;

(4) 设定绑定关键字，在交换器和队列之间建立好绑定关系;

(5) 发布者将网络消息投递到交换器，交换器根据消息的路由关键字和已经设置好的绑定关键字，将消息投递到一个或多个队列中。

3.1.3　AMQP 服务器

由交换器和消息队列两部分组成，主要功能是完成消息的路由和存储;

本文应用于发布/订阅模式，所以选择 Topic 类型的交换器，工作模型如图 11 所示，它可以根据不同的路由关键字和绑定关键字进行消息过滤和转发，把不同的消息路由到不同的队列中。

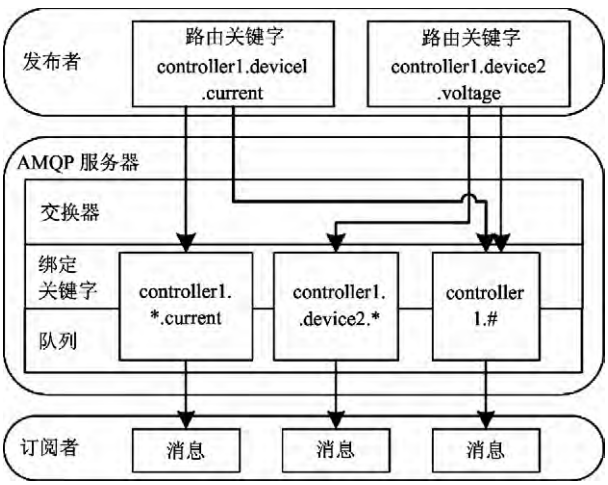

图 11　Topic 交换器工作模型

路由关键字和绑定关键字可以自行设定，使用有实际物理含义的信息表示更容易理解，假定发布者 1 的路由关键字设置为控制器 1. 设备 1. 电流值(controller1. device1. current) ，发布者 2 的路由关键字设置为控制器 1. 设备 2. 电压值(controller1. device2. voltage) 。AMQP 服务器中设置了三个消息队列，绑定关键字分别设置为 queue1: controller1. * . current、queue2: controller1. device2. * 、queue3: controller1. #(* 表示匹配一个标识符，#表示匹配 0 个或多个标识符) ，根据路由关键字和绑定关键字的匹配，发布者 1 将消息发送给 queue1 和 queue3，发布者 2 将消息发送给 queue2 和 queue3。

3.1.4　订阅者模块

订阅者模块首先需要完成消息接收的功能，连接到 AMQP 服务器中的队列，接收自身感兴趣的消息，获取队列中的网络消息后，进行消息处理过程，根据字段信息解码网络消息，获取数据信息集。订阅者的工作流程如下:

(1) 创建一个 channel，连接到 AMQP 服务器端;

(2) 创建一个消息队列，设置相关属性;

(3) 订阅者接收此消息队列中的网络消息并进行字段信息分析，解码网络消息。

3.2　功能测试

为了验证 OPC UA 发布/订阅模式中消息传输的正确性，本文编写了功能测试模块。每次测试设定不同的消息路由关键字和队列绑定关键字，记录发布者发出的消息是否到达正确的队列。测试结果如表 2 所示。

表 2　功能测试结果

路由关键字	绑定关键字		
	Controller1. * . current	Controller1. device2. *	controller1. #
controller1. device1. current	全部接收	不接收	全部接收
controller1. device2. voltage	不接收	全部接收	全部接收

测试设定每次传输的消息数量为 50 万个，并且进行了多次重复实验以提高结果数据的真实有效

性。当发布者将路由关键字设为 controller1. device1. current 时，绑定关键字为 controller1. * . current 和 controller1. #的队列可以正确接收 50 万个数据，与路由关键字和绑定关键字的匹配结果相同。同理，当发布者将路由关键字设为 controller1. device2. voltage 时，绑定关键字为 controller1. device2. * 和 controller1. #的队列可以正确接收 50 万个数据。测试结果表明，此模式具有较高的正确性和稳定性。

3.3 性能测试

考虑到工业现场存在数据量大的特点，本文通过编写性能测试模块，针对发布者发送不同数据大小的传输所需时间进行了比较，并且同时比较了订阅者断开状态，一个订阅者连接状态和多个订阅者连接状态对传输时间的影响，结果数据如表 3 所示。测试设定每次传输的消息数量为 50 万个。

表 3 性能测试结果数据

消息大小（字节）	速率（个/s）		
	断开状态	连接状态（1 个）	连接状态（4 个）
1000	3171	2792	2025
2000	2316	2094	1575
3000	1934	1749	1358
4000	1601	1407	1212
7000	1049	1011	812
10000	786	756	554

图 12 为传输时间对比图，曲线 1、曲线 2 和曲线3分别代表订阅者断开状态，一个订阅者连接状态和多个订阅者连接状态三种情况。由结果分析可知，发布者发送相同大小的消息数量时，订阅者断开连接状态所需的传输时间更少，传输速率更快。订阅者连接数量越多，发布者的传输速率越慢，但是目前的刷新率仍然可以达到 1 ~ 2ms，满足绝大多数工业需求。

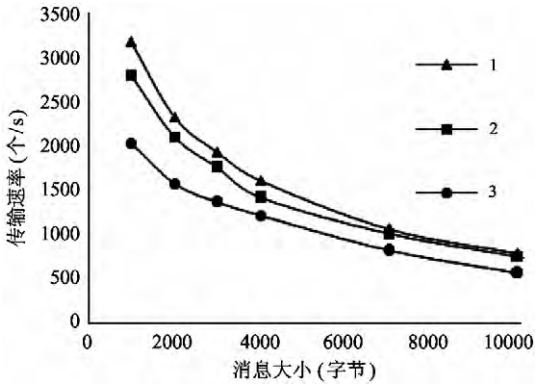

图 12 传输时间对比

4 结 论

本文将消息代理机制应用到 OPC UA 中，实现了 OPC UA 发布/订阅通信模式。此种模式中，发布者不用关心具体的节点信息发送给了哪些订阅者，而只需给节点信息设定好关键字，由消息代理自动将信息推送到相关联的订阅者。订阅者可自主订阅并随时上线接收消息。

OPC UA 发布/订阅通信模式适应于分布式系统的发展要求，在未来，可以将 OPC UA 发布/订阅和时间敏感网络（time-sensitive networking，TSN）相结合，进一步达到工业实时性和准确性的要求。

参考文献

[1] OPC Foundation. BS EN 62541-1 OPC Unified Architecture Specification Part1: Overview and Concepts 1. 04 [S]，2017. 07

[2] Cavalieri S，Stefano D D，Salafia M G，et al. Integration of OPC UA into a web-based platform to enhance interoperability [C]. In: Proceedings of the IEEE International Symposium on Industrial Electronics，Edinburgh，UK，2017. 1206-1211

[3] 王锴，郭海丰，徐皑冬，等. 面向智能制造的电磁线圈绝缘退化监测方法 [J]. 信息与控制，2017，46(4)：469-473

[4] 王华，刘枫. OPC UA 技术及应用 [J]. 工业控制计算机，2008，21(12)：38-39

[5] 苏延召，李艾华. 基于 OPC UA 的自动化系统集成技术研究 [J]. 测控技术，2011，30(3)：68-71

[6] OPC Foundation. BS EN 62541-14 OPC Unified Architecture Specification Part 14: PubSub Release Candidate1. 04. 24 [S]，2017. 02

[7] OPC Foundation. BS EN 62541-3 OPCUnified Architecture Specification Part 3: Address Space Model 1. 04

[S], 2017.07

[8] 陆会明, 阎志峰. OPC UA 服务器地址空间关键技术研究与开发[J]. 电力自动化设备, 2010, 30(7): 109-113

[9] OPC Foundation. BS EN 62541-6 OPCUnified Architecture Specification Part 6: Mappings 1.03 [S], 2015.07

[10] Kim W, Sung M. OPC-UA Communication Framework for PLC-based Industrial IoT Applications: Poster Abstract [C]. In: Proceedings of the IEEE/ACM 2nd International Conference on Internet-of-Things Design and Implementation, Pittsburgh, USA, 2017. 327-328

[11] 高晓婷. 基于 AMQP 的信息发布与订阅[硕士学位论文][D]. 浙江: 浙江工业大学信息工程学院, 2013. 6-8

[12] 吴炜鑫, 王宇, 王兴伟. 基于 AMQP 的校园消息总线系统的设计与实现[J]. 通信学报, 2013(s2): 180-188

Research and implementation of OPC UA Publish/Subscribe mode based on message broker

Liu Yang[*,**], Liu Mingzhe[*], Xu Aidong[*], Wang Kai[*], Han Xiaojia[*,**], Zhang Ning[*,**], TengYukun[*,**], Jiang Junhui[***]

([*] Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016)

([**] University of Chinese Academy of Sciences, Beijing 100049)

([***] Department of Electronic Systems Engineering, Hanyang University, Ansan 15588 Korea)

Abstract

In order to solve the problem of tight coupling and server performance bottlenecks existing in the OPC UA client/server communication mode, the overallarchitecture of OPC UA publish/subscribe communication mode is studied. This study analyzes the core functions of UA, including address space technology and data coding technology, and then, completes the function of publisher and subscriber using message broker. The experiment also verifies the stability of function and the data transmission capacity, the results show that this model can meet most of industrial needs.

Key words: OPC unified architecture, address space, data coding, publish/subscribe, message broker