

ANÁLISE COMPARATIVA DE IMPLEMENTAÇÕES DE SOFTWARE DE CÓDIGO ABERTO DO OPC UA

PABLO. F. S. MELO, EDUARDO. P. GODOY

Universidade Estadual Paulista (Unesp), Sorocaba, SP, Brasil

E-mails: pablofsmelo@gmail.com, epgodoy@sorocaba.unesp.br

Abstract— The OPC UA protocol is an open source technology defined to implement a communication layer in terms of Industrial Internet of Things and Industry 4.0. Currently, there are several open source and commercial software implementations of OPC UA organized from different programming languages, licenses and resources. This study presents an overview about OPC UA and OPC UA open source implementations which provide support for client and server implementations. In addition, a comparative analysis among these OPC UA implementations is presented, describing their characteristics and emphasizing the following topics: code version, mapping of technologies, security policies and available service sets.

Keywords— OPC UA, Open Source, Industrial Internet of Things, Industry 4.0, RAMI 4.0.

Resumo— O protocolo OPC UA é uma tecnologia de código aberto definida para implementar a camada de comunicação em termos de Internet das Coisas Industrial e Indústria 4.0. Atualmente, existem diversas implementações de software de código aberto e comerciais do OPC UA, estruturadas a partir de diferentes linguagens de programação, licenças e recursos. Este estudo apresenta uma revisão sobre o OPC UA e sobre as principais implementações de código aberto do OPC UA que fornecem suporte para o desenvolvimento de clientes e servidores. Adicionalmente, uma análise comparativa entre essas implementações do OPC UA é apresentada, descrevendo suas características e enfatizando os seguintes tópicos: versão do código, mapeamentos de tecnologias, políticas de segurança e conjuntos de serviços disponibilizados.

Palavras-chave— OPC UA, Código Aberto, Internet das Coisas Industrial, Indústria 4.0, RAMI4.0.

1 Introdução

A Internet das Coisas Industrial (IIoT - *Industrial Internet of Things*) objetiva principalmente a transferência de informações, análise de dados e aplicações através da utilização de tecnologias que afetam diretamente a automação em infraestrutura e melhoram a produtividade empresarial (Hassanzadeh et al., 2015), tornando-se um dos principais elementos da Indústria 4.0 (I4.0). Eruvankai et al. (2017) afirmam que apesar do grande potencial da IIoT, há desafios quanto à sua adoção, principalmente no que diz respeito aos seguintes tópicos: segurança de dados, confiabilidade de comunicação, interoperabilidade e escalabilidade. Os autores afirmam que para superar estes desafios tem-se o protocolo OPC UA (*Open Platform Communications Unified Architecture* – Plataforma de Comunicação Aberta de Arquitetura Unificada). O OPC UA apresenta uma arquitetura segura e totalmente escalável, sendo esta, baseada no conceito de SOA (*Service Oriented Architecture* – Arquitetura Orientada a Serviços).

Dessa maneira, o OPC UA se estabelece como sendo um padrão essencial para aplicações de IIoT (Wollschlaeger, Sauter & Jasperneite, 2017). Hoppe (2017) afirma que não há possibilidade de aplicações de Indústria 4.0 sem o OPC UA, visto que o Modelo de Arquitetura de Referência para a Indústria 4.0 (RAMI 4.0) recomendou apenas o OPC UA para implementação de sua camada de comunicação.

Além de atender os requisitos da I4.0 e IIoT, o protocolo OPC UA originalmente foi criado para solucionar limitações encontradas nas especificações do seu antecessor, o OPC Clássico (DA). É possível

considerar que o OPC UA unifica todas as especificações do OPC Clássico e as leva para o estado da arte usando SOA (Lehnhoff et al., 2011). Dentre os diversos avanços fornecidos pelo OPC UA tem-se a possibilidade de desenvolvimento empregando múltiplas plataformas, como Windows, Linux e sistemas embarcados. Este foi viabilizado devido à OPC Foundation (entidade responsável pelas especificações do OPC UA) disponibilizar *SDKs* (*Software Development Kit* - Kits de Desenvolvimento de Software) estruturados em diversas linguagens de programação, como C/C++, C# e Java (Huiming & Zhifeng, 2010).

Da mesma forma, a difusão do OPC UA também foi norteada para disponibilização de implementações de softwares de código aberto desse protocolo pela *OPC Foundation*. Hoppe e Burke (2016) destacam que a intenção dessa disponibilização é tornar o OPC UA como solução para a Internet de Tudo, incluindo IIoT, IoT e M2M (*Machine to Machine* – Máquina para Máquina). Nesse contexto, Palm et al. (2015) discutem as vantagens dos projetos de código aberto para fomentar aplicações da IIoT e da I4.0. Adicionalmente, os autores analisam as vantagens de usar uma implementação de código aberto dentro do OPC UA e citam algumas implementações de código aberto atualmente existentes.

É possível encontrar estudos que comparam implementações de software do OPC UA. Profanter et al. (2017) avaliam implementações de código aberto desenvolvidas nas linguagens C/C++ e Java, com o propósito de verificar quais suportam o recurso de LDS-ME (*Local Discovery Server with Multicast Extension* – Servidor de Descoberta Local com Extensão Multicast). Haskamp et al. (2017) apresentam uma visão geral das soluções existentes do OPC UA

e mostram resultados de uma análise comparativa inicial, baseada em um conjunto de critérios definidos pelos próprios autores. Nsiah et al. (2017) comparam três implementações de código aberto para uma aplicação de mapeamento e integração de dispositivos com base em EDDL (*Electronic Device Description Language* – Linguagem de Descrição de Dispositivo Eletrônico), juntamente com o OPC UA.

Este estudo foca na análise comparativa entre as principais implementações de código aberto do OPC UA que contemplam aplicativos de clientes e servidores. Por esse motivo, são descritos alguns aspectos construtivos gerais do OPC UA, considerando as principais funcionalidades básicas e após isso, são realizadas análises descritivas que avaliam as particularidades de cada uma das implementações citadas.

2 Visão geral do OPC UA

A arquitetura do OPC UA inclui basicamente dois tipos de aplicativos: *UA Client* (Cliente OPC UA) e *UA Server* (Servidor OPC UA). Criado em 2006, o OPC UA é um protocolo para comunicação industrial, padronizado na IEC 62541 (Gruner et al., 2016) e definido como um conjunto de especificações independente de uma determinada plataforma, sendo que diversos tipos de dispositivos e sistemas podem se comunicar por meio de mensagens trocadas entre clientes e servidores nos mais variados tipos de redes, mantendo assim, uma arquitetura baseada em padrões de tecnologias Web. Em função disso, o protocolo busca assegurar a interoperabilidade entre seus componentes (Rohjans et al., 2010) e apresenta como principal característica, a definição de poderosos modelos de informações que organizam os dados dentro do espaço de endereço (*AddressSpace*) de um servidor OPC UA em termos de estrutura e semântica. Tal fator caracteriza o padrão como sendo uma perfeita escolha para integração vertical dos sistemas de automação (Grossmann et al., 2008).

O protocolo é estruturado através de 14 especificações, distribuídas em 3 partes, sendo elas: núcleo (*core*), tipos de acesso (*access type*) e utilidades (*utility*), conforme demonstrado na Figura 1 (OPC Foundation, 2017).

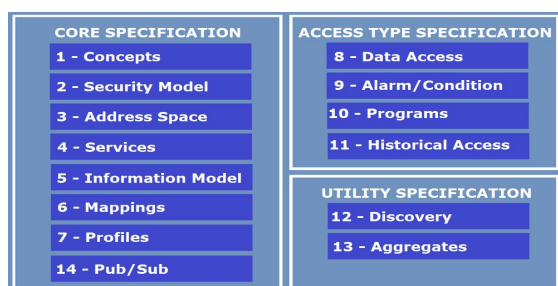


Figura 1. Conjunto de especificações do OPC UA.

Com base nesse conjunto de especificações, as seguintes funcionalidades básicas foram definidas para a implementação do protocolo OPC UA: Mapeamento de Tecnologias e Serviços.

2.1 Mapeamento de tecnologias

O mapeamento de tecnologias pode ser definido em *UA Web Services* ou *UA Native* (Leitner & Mahnke, 2006). O mapeamento *UA Web Services* utiliza algumas especificações de *Web Services*, enquanto o *UA Native* implementa uma simples rede binária e integra mecanismos de segurança. Os principais elementos que compõem o mapeamento de tecnologias são: codificação dos dados, transporte e segurança (OPC Foundation, 2017). Tais elementos são fundamentais para a criação da pilha de software do OPC UA e alguns aspectos das camadas que constituem o mapeamento de tecnologias são relatados:

a) Codificação: responsável pela codificação dos dados entre clientes e servidores. Essa camada pode ser feita em XML (*XML Extensible Markup Language*) ou UA Binário (Leitner e Mahnke, 2006).

b) Transporte: responsável pelo transporte dos dados entre clientes e servidores. Em se tratando da camada de transporte, o *mapeamento UA Web Services* emprega o SOAP (*Simple Object Access Protocol*)/HTTPS (*Hyper Text Transfer Protocol Secure*), enquanto o *UA Native* executa diretamente no *UA TCP/IP* (*Transmission Control Protocol/Internet Protocol*).

c) Segurança: a camada de segurança é imprescindível para execução de uma comunicação segura, sendo responsável por assinar, codificar e decodificar as mensagens trocadas entre clientes e servidores OPC UA. A especificação de número 2 descreve o modelo de segurança, incluindo diferentes mecanismos e parâmetros de segurança.

Conforme Haskamp et al. (2017), antes de abrir um canal seguro, o servidor e o cliente devem concordar com o mecanismo de segurança usado para a sessão de comunicação. Para isso, são definidas 4 políticas de segurança para encriptação dos dados: **None** - sem configuração de segurança, **Basic128Rsa15** - configurações médias de segurança, **Basic256** - configurações médias para altas de segurança; **Basic256Sha256**: configurações altas de segurança. Outro ponto a ser considerado, diz respeito a Autenticação. Para a autenticação, a especificação de número 7 disponibiliza as seguintes formas para trocar *tokens* de identidade do usuário (OPC Foundation, 2017): **Anonymous** - não é necessária nenhuma identidade de usuário, **User Name Password**: combinações de nomes e senhas para identificar usuários, **X.509 Certificate**: par de chaves público/privado para identidade de usuário.

2.2 Serviços

Os conjuntos de serviços do OPC UA são definidos na especificação de número 4, disponibilizada pela OPC Foundation. Cada um destes conjuntos

define um grupo lógico usado para acessar um aspecto particular do servidor OPC UA. Os principais serviços do OPC UA são: Descoberta (*Discovery*), Canal de Segurança (*SecureChannel*), Sessão (*Session*), Gerenciamento de Nó (*NodeManagement*), Visualização (*View*), Pergunta (*Query*), Atributo (*Attribute*), Método (*Method*), Item Monitorado (*MonitoredItem*) e Assinatura (*Subscription*).

3 Análise Descritiva

Considerando as implementações de código aberto do OPC UA que integram tanto aplicativos de clientes como de servidores, existem oito projetos utilizando diferentes linguagens de programação, licenças e fornecimento de recursos para tais aplicativos (Profanter, 2017a). Neste trabalho é feita uma revisão e explanadas as principais características de cada implementação, apresentando em alguns casos, exemplos de aplicação desses projetos.

3.1 node-opcua

O projeto node-opcua consiste de uma implementação do OPC UA escrita em *Javascript* e *Node.js*. A escolha pelo *Node.js* fez-se por esta linguagem ser uma excelente estrutura para projetar aplicativos assíncronos. O node-opcua pode ser executado em todas as plataformas que suportam *Node.js*, como Linux, Mac e Windows (Rossignon, 2017). Em termos de licença, o node-opcua utiliza a licença MIT (*Massachusetts Institute of Technology*) e atualmente encontra-se na versão 0.0.64.

A documentação da pilha conta com inúmeros testes funcionais e um conjunto de exemplos práticos para ajudar o programador a estruturar suas aplicações OPC UA. Um exemplo disso são as possibilidades de criação de clientes e servidores OPC UA disponíveis em tutoriais que abordam desde a preparação até a execução e testes das aplicações.

Outro ponto a ser considerado é utilização do node-opcua em sistemas embarcados. No site oficial está relatado que o projeto vem sendo otimizado para implementações em *Raspberry Pi*, permitindo assim a integração do servidor OPC UA com sensores e dispositivos de entradas e saídas (Rossignon, 2017). Derivando-se do projeto node-opcua, existem alguns repositórios que tratam diferentes tópicos, como o node-opcua-isa95, sendo uma extensão da ISA95 para um servidor OPC UA, e o opcua-commander que fornece um CLI (*Command Line Interface* - Interface de Linha de Comando) para um OPC UA Cliente.

A respeito da utilização do node-opcua, Baumann et al. (2017) desenvolvem uma aplicação do OPC UA que controla uma impressora 3D. Fleischmann et al. (2017) desenvolvem um conceito de uma IHM (Interface Homem Máquina) configurável e adaptável que monitora plantas discretas de produção, com base nas tecnologias OPC UA e *Automa-*

tionML. Rúnarsson (2016) propõe uma alternativa de CLP (Controlador Lógico Programável) de hardware e software de código aberto usando o node-opcua para criação do servidor da aplicação.

3.2 open62541

O open62541 é uma implementação de código aberto do OPC UA escrito em linguagem C/C++ e licenciado sobre o *Mozilla Public License* v2.0. A biblioteca do open62541 é utilizável com boa parte dos principais compiladores, além de fornecer as ferramentas necessárias para criação de aplicações de clientes e servidores OPC UA (Profanter, 2017b).

O open62541 é escalável, suportando arquitetura *Multi-Threaded*, onde cada conexão ou sessão pode ser operada por threads individuais (Kozar, 2016). Destaca-se que o open62541 é portátil, pois suporta POSIX (*Portable Operating System Interface* - Portável entre Sistemas Operacionais), o que torna o projeto habilitado em diferentes plataformas, como, Windows, Linux, MacOS, Android e Sistemas Embarcados.

O open62541 está na sua terceira versão (v0.3) e disponibiliza uma extensa e excelente documentação para suportar a criação de clientes e servidores OPC UA. Outra característica é de que nesta versão não foram encontrados erros que podem ser indicados através das ferramentas oficiais de testes de conformidade (CTT) (Profanter, 2017b).

A respeito da utilização do open62541, Contrenas, Garcia e Pastrana (2017) expõem as características essenciais que permitem que um sistema de fabricação seja adaptado para se tornar um aplicativo da I4.0. Neste trabalho a parte de comunicação é implementada usando o OPC UA. Johansson (2017) discute como uma simulação interativa de modelos de sistemas físicos para o *OpenModelica* pode ser projetada usando um cliente OPC UA. Muller, Wings e Bergmann (2017) apresentam um servidor OPC UA funcionando através de uma placa Arduino Yun.

3.3 Eclipse Milo

O Eclipse Milo foi apresentado durante a *EclipseCon Europe* em 2016 (Eclipse, 2016). O Eclipse Milo, parte do *Eclipse IoT Project*, é uma implementação de código aberto do OPC UA baseado na linguagem de programação Java. O projeto é desenvolvido pela Fundação Eclipse que fornece uma boa documentação é licenciado sob o EPL -1.0 (*Eclipse Public License – Version 1.0*). O Eclipse Milo oferece uma pilha e SDK, onde há um núcleo de código comum compartilhado entre cliente e servidor (Reimann, 2017).

A respeito da utilização do Eclipse Milo, Dorofeev et al. (2017) apresentam o conceito de um adaptador de dispositivos que permite que o dispositivo seja facilmente conectado à sistemas de produção usando o OPC UA. Profanter et al. (2017) avalia algumas implementações de código aberto e certificam

a Eclipse Milo como uma escolha perfeita para um sistema de produção multinível *Plug & Produce*.

3.4 ASNeg

O ASNeg (*Automation Service Next Generation* – Serviço de Automação de Próxima Geração) é uma implementação de código aberto do OPC UA desenvolvido em C++ e licenciado sob o Apache 2.0. Além de fornecer soluções relacionadas a clientes e servidores OPC UA, o ASNeg conta com 9 repositórios para outras aplicações, incluindo OPCUADB (Base de dados do Servidor), OPCUaWebServer e OPCU-ARaspberry (Huebl, 2018).

3.5 UA.NET

Desenvolvido diretamente por membros da *OPC Foundation*, o UA.NET é uma implementação de código aberto do OPC UA, desenvolvido em C# usando o .NET Standard Library (Moldoven, Barnstedt, Oursel, 2017). O UA.NET fornece uma pilha OPC UA contendo uma série de aplicativos de amostras. O UA.NET é licenciado sobre o GLP (*GNU General Public License*) e RCL (*Reciprocal Community License*). O RCL permite que os membros da *OPC Foundation* implementem suas aplicações usando a pilha UA.NET sem que seja obrigatório divulgar o código do aplicativo. Enquanto isso, os não membros devem divulgar o código de suas aplicações ao usar a pilha UA.NET (OPC Foundation, 2016).

O .Net Standard Library permite que você desenvolva aplicativos que funcionem em todas as plataformas comuns disponíveis, incluindo Linux, iOS, Android (via Xamarin) e Windows 7/8/8.1/10 (incluindo edições embarcadas/IoT), sem requerer modificações específicas da plataforma. Além disso, também são suportados aplicativos e serviços em nuvem (como ASP.Net, DNX, Azure Web, Azure Webjobs, Azure Nano Server e Azure Service Fabric) (Moldoven & Barnstedt & Oursel, 2017).

3.6 FreeOPCUA

O FreeOPCUA é composto por um conjunto de bibliotecas de código aberto do OPC UA, construídas em C++ e Python. O FreeOPCUA conta com uma biblioteca LGPL (*GNU Lesser General Public License*) C++ para desenvolvimento de aplicações de clientes e servidores OPC UA, utilizando a linguagem de programação C++ que é denominada “freeopcua”. Outra possibilidade é pela aplicação da biblioteca escrita em *Python*, fornecida através do repositório no GitHub, chamado “phyton-opcua” (Rykonanov, 2017).

Além disso, o FreeOPCUA fornece duas soluções de GUI (*Graphical User Interface*): opcua-client-gui e opcua-modeler. A primeira implementa as funcionalidades básicas, incluindo a alteração de dados e eventos, escrita de valores nas variáveis, listagem de atributos e referências. A segunda é usada para projetar espaços de endereços usando XML, o que permite que o XML produzido seja portátil para qualquer SDK (Rykonanov, 2017).

A respeito da utilização do FreeOPCUA. Montavon, Peterek e Schmitt (2017) usam o OPC UA num modelo proposto para implantar o conceito de metrologia para a Indústria 4.0. Åkerman, Berglund e Ekered (2016) apresentam soluções que exemplificam a integração vertical e horizontal de sistemas usando OPC UA. Dentro dessas soluções é implementado um sistema RFID combinado com a implementação FreeOPCUA.

4 Análise Comparativa

Perante os principais aspectos do OPC UA apresentados, este tópico retrata os resultados obtidos através de uma análise comparativa entre as principais implementações de código aberto do OPC UA (Tabela 1) enfatizando os seguintes tópicos: versão do código, mapeamentos de tecnologias, políticas de segurança, autenticação e conjuntos de serviços.

Tabela 1. Análise Comparativa entre as principais implementações de software de código aberto do OPC UA.

Parâmetros		node-opcua	open62541	Eclipse Milo	ASNeg	UA.NET	FreeOPCUA (Python)	FreeOPCUA (C++)
Versão do código		v0.2.2	v0.3	v0.2.1	v1.6.2	-	v0.90.6	v0.2.2
Mapeamentos de Tecnologias	UA-TCP UA-SC UA Binary	✓	✓	✓	✓	✓	✓	✓
	HTTP WS-SC UA Binary			✓		✓		
	HTTP WS-SC UA XML			✓		✓		
Políticas de Segurança	Nenhuma	✓	✓	✓	✓	✓	✓	✓
	Basic128Rsa15	✓	✓	✓	✓	✓	✓	
	Basic256	✓		✓	✓	✓	✓	
	Basic256Sha256	✓		✓	✓	✓		
Autenticação	Anonymous	✓	✓	✓	✓	✓	✓	✓
	User Name Password	✓	✓	✓		✓	✓	
	X509 Certificate			✓		✓	✓	
Serviços de Descoberta	FindServers	✓	✓	✓	✓	✓	✓	✓
	GetEndpoints	✓	✓	✓	✓	✓	✓	✓

	RegisterServer	✓	✓	✓	✓	✓	✓	✓
Serviços de Canal de Segurança	OpenSecureChannel	✓	✓	✓		✓	✓	✓
	CloseSecureChannel	✓	✓	✓		✓	✓	✓
Serviços de Sessão	CreateSession	✓	✓	✓	✓	✓	✓	✓
	CloseSession	✓	✓	✓		✓	✓	✓
	ActivateSession	✓	✓	✓	✓	✓	✓	✓
	Cancel			✓	✓	✓	✓	✓
Serviços de Visualização	Browse	✓	✓	✓	✓	✓	✓	✓
	BrowseNext	✓	✓	✓	✓	✓	✓	✓
	RegisterNodes	✓	✓	✓	✓	✓	✓	✓
	UnregisterNodes	✓	✓	✓	✓	✓	✓	✓
Serviços de Atributos	Read	✓	✓	✓	✓	✓	✓	✓
	Write	✓	✓	✓	✓	✓	✓	✓
	HistoryRead			✓	✓	✓	✓	
	HistoryUpdate			✓	✓	✓	✓	✓
Serviços de Itens Monitorados	CreateMonitoredItems	✓	✓	✓	✓	✓	✓	✓
	ModifyMonitoredItems	✓	✓	✓	✓	✓	✓	
	SetMonitoringMode	✓	✓	✓	✓	✓	✓	
	SetTriggering			✓	✓	✓	✓	
	DeleteMonitoredItems	✓		✓	✓	✓	✓	✓
Serviços de Assinatura	CreateSubscription	✓	✓	✓	✓	✓	✓	✓
	ModifySubscription	✓	✓	✓	✓	✓	✓	✓
	DeleteSubscriptions	✓	✓	✓	✓	✓	✓	✓
	Publish	✓	✓	✓	✓	✓	✓	✓
	Republish	✓	✓	✓	✓	✓	✓	✓
Serviços de Gerenciamento de Nós	AddNodes		✓	✓	✓	✓	✓	✓
	AddReferences		✓	✓	✓	✓	✓	✓
	DeleteNodes		✓	✓	✓	✓	✓	✓
	DeleteReferences		✓	✓	✓	✓	✓	
Serviços de Perguntas	QueryFirst			✓	✓	✓	✓	
	QueryNext			✓	✓	✓	✓	

5 Conclusão

O OPC UA é um padrão extremamente importante dentro do contexto industrial, pois além de integrar as camadas que vão desde o chão de fábrica até os sistemas corporativos, é também, o único padrão definido pelo RAMI 4.0 para implementar a camada de comunicação em termos de I4.0 e aplicações de IIoT.

Este trabalho apresentou uma revisão do protocolo OPC UA, bem como um detalhamento das funcionalidades básicas, mapeamento de tecnologias e serviços do OPC UA. Com relação às análises desenvolvidas ao longo deste estudo, concluiu-se que todas as implementações viabilizam a comunicação OPC UA e se encontram em estágio de desenvolvimento, porém apenas a Eclipse Milo e UA.NET apresentam todos os requisitos estabelecidos.

Referências Bibliográficas

- Baumann, F. W; Falkenthal, M; Hudert, S; Odefey, U and Zimmermann, M (2017). Cyber- Physical Sytem Control via Industrial Protocol OPC UA. Institute of Architecture of Application Systems, Universitat Stuttgart, Germany.
- Contrenas, J.D; Garcia, J.I and Pastrana, J.D (2017). Developing of Industry 4.0 Applications. International Journal of Online Engineering, iJOE – Vol. 13, No. 10, 2017.
- Dorofeev, K; Cheng, C.H; Guedes, M.; Ferreira, S; Profanter, S and Zoitl, A (2017). Device adapter concept towards enabling plug&produce production environments. *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation*, Limassol, pp. 1-8.
- Eclipse (2016). Eclipse Milo 0.1.0 Release Review. Acesso Janeiro, 2018.Disponível em: <<https://projects.eclipse.org/projects/iot.milo/reviews/0.1.0-release-review>> .

- Eruvankai, S; Muthukrishnan, M and Mysore, A, K (2017). Accelerating IIoT Adoption with OPC UA. In: Internetworking Indonesia Journal, v.9, n.1, p.3-8.
- Fleischmann, H; Brossog, M; Beck, M and Franke, J (2017). Automated Generation of Human-Machine Interfaces in Electric Drives Manufacturing. IEEE Institute for Factory Automation and Production Systems.
- Grossmann, D; Bender, K and Danzer, B (2008). OPC UA based Field Device Integration. In SICE Annual Conference. 2008 IEEE International Conference on. p.933-938.
- Gruner, S; Pfrommer, J and Palm, F (2016). RESTful Industrial Communication with OPC UA. Industrial Informatics, IEEE Transactions on, v.12, n.5, p.1832-1841.
- Haskamp, M; Meyer, R; Mollmann, F; Orth, F and Colombo, A.W (2017). Benchmarking of existing OPC UA implementations for Industrie 4.0-compliant digitalization solutions. 2017 IEEE 15th International Conference on Industrial Informatics, Emden, pp. 589-594.
- Hassanzadeh, A; Modi, S and Mulchandani, S (2015). Towards Effective Security Control Assignment in the Industrial Internet of Things. 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, p.795-800.
- Hoppe, S (2017). There is no Industrie 4.0 without OPC UA. Disponível em: <http://opcconnect.opcfoundation.org/2017/06/the-re-is-no-industrie-4-0-without-opc-ua/>. Acesso: Dezembro, 2017.
- Hopper, S and Burke, T (2017). OPC Foundation Announces OPC UA Open Source Availability. Disponível em: <https://opcfoundation.org/news/opc-foundation-news/opc-foundation-announces-opc-ua-open-source-availability/>. Acesso: Janeiro, 2017.
- Huebl, K (2017). Willkommen beim ASNeG Git Repository Acesso: Janeiro, 2018. Disponível em: <https://81.169.197.52:8443/repositories/>.
- Huiming, L and Zhifeng, Y (2010). Research on Key Technology of the Address Space for OPC UA Server. In 2nd International Conference on Advanced Computer Control. 2010 IEEE International Conference on. IEEE, p.278-281.
- Johansson, C.F (2017). Open Modelica Interactive Simulation using an OPC UA client. Dissertação Linköping University, Suécia.
- Kozar, S (2016). Integration of IEC 61499 with OPC UA. Dissertação Czech Technical University in Prague.
- Lehnhoff, S; Mahnke, W and Rohjans, S (2011). IEC 61850 based OPC UA Communication- The Future of Smart Grid. In 17th Power Systems Computation Conference, Stockholm Sweden, p.1-9.
- Leitner, S.H and Mahnke, W (2006). OPC UA – Service-oriented Architecture for Industrial Applications. In ABB Research Center.
- Moldoven, A; Barnsted, E and Oursel, E (2017). OPC UA .Net Standard Stack and Samples. Disponível em: <https://github.com/OPCFoundation/UA-.NETStandard>. Acesso: Fevereiro, 2018.
- Montavon, B; Peterek, M and Schmitt, R. H (2017). Communication Architecture for Multiple Distributed Large Volume Metrology Systems. 2017 IEEE International Systems Engineering Symposium (ISSE).
- Muller, M; Wings, E and Bergmann, L (2017). Developing open source cyber-physical systems for service-oriented architectures using OPC UA. 2017 IEEE 15th International Conference on Industrial Informatics, Emden, pp. 83-88;
- Nsiah, A.K; Schappacher, M and Sikora, A. Dynamic mapping of EDDL device descriptions to OPC UA. 2nd International Conference on Measurement Instrumentation and Electronics. Journal of Physics: Conference Series, 2017
- OPC Foundation (2016). Build OPC UA.NET applications using .NET Standard Library, 2016. Disponível em: <http://opcfoundation.github.io/UA-.NETStandard/>. Acesso: Janeiro, 2018.
- OPC Foundation (2017). Unified Architecture: OPC Unified Architecture Specification. Disponível em: <https://opcfoundation.org/developer-tools/specifications-unified-architecture>. Acesso: Janeiro, 2018.
- Palm, F; Gruner, S; Pfrommer, J; Graube, M and Urbas, L (2015). Open Source as Enabler for OPC UA in Industrial Automation. 2015 IEEE 20th Conference Technologies & Factory Automation (ETFA), Luxembourg, pp, 1-6.
- Palomen, O (2010). Object Oriented Implementation of OPC UA Information Models in Java. 2010. 81 f. Dissertação Aalto University – School of Electrical Engineering. Espoo.
- Profanter, S; Dorofeev, K.L.; Zoitl, A and Knoll, A (2017). OPC UA for Plug & Produce: Automatic Device Discovery using LDS-ME. 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, Cyprus, pp. 1-8.
- Profanter, S (2017a). Open 62541. Disponível em: <https://github.com/open62541/open62541/wiki/List-of-Open-Source-OPC-UA-Implementations>. Acesso: Novembro, 2017.
- Profanter, S (2017b). Open62541. Disponível em: <https://github.com/open62541/open62541>. Acesso: Janeiro, 2018.
- Reimann, J (2017). OPC UA solutions with Eclipse Milo. Disponível em: <https://dentrassi.de/2017/09/14/creating-opc-ua-solutions-eclipse-milo/>. Acesso: Janeiro, 2018.
- Rohjans, S; Uslar, M; Appelrath, J (2010). OPC UA and CIM: Semantics for the Smart Grid. In IEEE PES T&D. 2010 IEEE Conference on, p.1-8.
- Rossignon, E. (2017). Node-opcua: an implementation of a OPC UA stack fully written in javascript

and nodejs. Disponível em:
<https://github.com/node-opcua/node-opcua>.
Acesso: Setembro, 2017.

Rúnarsson, S (2017). Open Source Hardware and Software Alternative to Industrial PLC. Dissertação University College of Southeast Norway, Noruega.

Rykonanov, A (2017). FreeOPCUA Library. Disponível em: <https://github.com/FreeOpcUa>. Acesso: Fevereiro, 2018.

Wollschlaeger, M; Sauter, T and Jasperneite, J (2017). The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0, IEEE Industrial Electronics Magazine, vol.11, pp.17-2017.