



GEOVANNE VICHIAATTO FIGUEIREDO

**UMA REVISÃO SOBRE A TECNOLOGIA OPC, SUAS
VARIACIONES E APLICAÇÕES**

LAVRAS – MG

2020

GEOVANNE VICHIAATTO FIGUEIREDO

**UMA REVISÃO SOBRE A TECNOLOGIA OPC, SUAS VARIAÇÕES E
APLICAÇÕES**

Trabalho de Conclusão de Curso
apresentado à Universidade Federal de
Lavras, como parte das exigências do
Curso de Engenharia de Controle e
Automação, para obtenção do título de
Bacharel.

Prof. Dr. Dimitri Campos Viana

Orientador

LAVRAS – MG

2020

GEOVANNE VICHIAATTO FIGUEIREDO

**UMA REVISÃO SOBRE A TECNOLOGIA OPC, SUAS VARIAÇÕES E
APLICAÇÕES**

A SURVEY ON OPC TECHNOLOGY, ITS VARIATIONS AND APPLICATIONS

Trabalho de Conclusão de Curso
apresentado à Universidade Federal de
Lavras, como parte das exigências do
Curso de Engenharia de Controle e
Automação, para obtenção do título de
Bacharel.

APROVADO em 15/06/2020.

Dr. Fábio Domingues de Jesus – UFLA

Dr. Felipe de Oliveira e Silva – UFLA

Prof. Dr. Dimitri Campos Viana

Orientador

LAVRAS – MG

2020

AGRADECIMENTOS

Agradeço, primeiramente, à Universidade Federal de Lavras pela oportunidade e pelo alto padrão de conhecimento ofertado aos alunos do curso de bacharelado em Engenharia de Controle e Automação.

Agradeço à minha mãe pelo apoio e amor incondicional durante a minha jornada. Agradeço também às minhas irmãs Beatriz, Giulia e Isabela pelo amor e carinho.

Agradeço ao tio Emerson e à tia Débora por terem me acolhido na cidade de Lavras e me oferecido sua casa como moradia e aconchego, com muito apoio durante os anos da minha graduação.

Agradeço à minha namorada Mayara, por todo amor, carinho, cuidado, apoio, companhia e paciência durante a graduação e por ser a melhor namorada que alguém pode ter.

Agradeço aos amigos e colegas, principalmente do PHD Esporte Clube, que estiveram junto comigo nessa caminhada.

Agradeço o professor Dimitri, pela orientação deste trabalho e pelo ensino nas disciplinas Informática Industrial e Informática Industrial Avançada.

MUITO OBRIGADO!

Resumo

Na maioria das plantas industriais existe um grande número de instrumentos de campo e de controladores, além de sistemas de supervisão e aquisição de dados, que devem trocar dados entre si, via rede. Historicamente, com o surgimento de uma grande quantidade de protocolos e meios de transporte, a complexidade para se estabelecer conexões confiáveis entre estes dispositivos é bastante significativa. Nesse contexto, o OPC (*Object Linking and Embedding for Process Control*, ou, *OLE for Process Control*, quando surgiu; e *Open Platform Communications*, na atualidade) foi desenvolvido para atender a demanda de um padrão de comunicação e interoperabilidade que simplificasse a descrita situação. Durante sua consolidação, a OPC Foundation foi criada para gerenciá-lo e esta publicou diversas revisões e variações do padrão original, permitindo, entre outras coisas, o acesso a dados presentes em Controladores Lógico Programáveis (CLPs), a dados armazenados em histórico e o disparo de notificações de alarmes e eventos gerados por sistemas de supervisão. Posteriormente, na primeira década dos anos 2000, buscando alta performance em redes locais, acesso externo por meio de *firewalls* e total independência com relação à tecnologia de terceiros, a OPC Foundation publicou uma nova especificação do padrão, denominada *OPC Unified Architecture* (OPC UA), unificando a interface das versões anteriores e aumentando a escalabilidade e interoperabilidade entre sistemas e dispositivos do chão de fábrica e de gestão empresarial. Perante a relevância do assunto para os profissionais que trabalham com sistemas de automação industrial, este trabalho tem como objetivo apresentar, em formato de um artigo de revisão bibliográfica, os conceitos fundamentais sobre a tecnologia OPC, levantando as motivações para seu surgimento e informações técnicas sobre o funcionamento das principais especificações publicadas pela OPC Foundation.

Palavras-chave: OPC Foundation; Interoperabilidade; Padrão de comunicação; Acesso a dados; OPC Clássico; OPC UA.

Abstract

Most industrial plants have a large number of field instruments and controllers, in addition to data acquisition and supervisory systems, which must exchange data with each other via network. Historically, with the emergence of a large number of protocols and bus systems, the complexity of establishing reliable connections between these devices is quite significant. In this context, OPC (Object Linking and Embedding for Process Control, or, OLE for Process Control originally; and Open Platform Communications, nowadays) was developed to meet the demand for a communication and interoperability standard that would simplify the aforementioned situation. During its consolidation, the OPC Foundation was created to manage the latter, and it published several revisions and variations of the original standard, allowing, among other things, access to data present in Programmable Logic Controllers (PLCs), data stored in history and triggering notifications of alarms and events generated by supervisory systems. Later, in the first decade of the 2000s, aiming high performance on local networks, external access through firewalls and total independence from third-party technologies, the OPC Foundation published a new specification of the standard, called Unified Architecture, unifying the interface of previous versions and increasing the scalability and interoperability between factory floor systems and devices and business management ones. Given the relevance of the subject for professionals working with industrial automation systems, this paper aims to present, in the format of a bibliographic review article, the fundamental concepts about OPC technology, raising the motivations for its emergence and technical information about the main specifications published by the OPC Foundation.

Keywords: OPC Foundation; Interoperability; Communication standard; Data access; Classic OPC; OPC UA

SUMÁRIO

PRIMEIRA PARTE	8
1 INTRODUÇÃO	9
2 CONSIDERAÇÕES FINAIS.....	11
REFERÊNCIAS	12
SEGUNDA PARTE	13

PRIMEIRA PARTE

1 INTRODUÇÃO

Em uma planta industrial pode existir um grande número de diferentes dispositivos e sistemas interligados via redes de dados (Schwarz e Börcsök, 2013). A presença de diversos Controladores Lógico Programáveis (CLPs), sistemas de Supervisão, Controle e Aquisição de Dados (SCADAs) e Painéis Eletrônicos de Operação (PEOs) de diferentes fabricantes, famílias e modelos tende a elevar a complexidade para se estabelecer comunicações confiáveis entre tais elementos. Isso se deve ao fato de que, ao longo do tempo, várias empresas como Siemens, Rockwell, Schneider e GE desenvolveram protocolos proprietários, que funcionavam sob diferentes meios de transporte de dados (Schwarz e Börcsök, 2013; Coelho *et al.*, 2016).

Nesse contexto, fez-se necessário o estudo e o desenvolvimento de soluções para comunicação e interoperabilidade entre dispositivos e sistemas de maneira independente de tecnologias fechadas (Coelho *et al.*, 2016). Dessa forma, em meados da década de 1990, o OPC foi desenvolvido por membros de empresas do ramo de controle e aquisição de dados como um padrão de comunicação o qual, atuando como um *driver* intermediário, permite o acesso a dados presentes na memória dos CLPs por sistemas SCADA e PEOs, independentemente de fornecedores (OPC Foundation, 2019; Visaya, 2020). O OPC alcançou grande relevância por ter se tornado um padrão internacional de comunicação e, desse modo, se tornou uma ferramenta de conhecimento imprescindível para engenheiros e pesquisadores da área de automação de processos industriais (Mahnke, Leitner e Damm, 2009).

Sendo assim, o presente trabalho busca apresentar, em sua segunda parte, os conceitos fundamentais acerca da tecnologia OPC, suas variações e aplicações. O artigo de revisão bibliográfica proposto tem como objetivo fornecer informações sobre o histórico do desenvolvimento do padrão OPC e seu princípio de funcionamento, levantando informações técnicas sobre as especificações publicadas pela OPC Foundation desde a década de 1990.

O artigo apresentado na segunda parte está organizado da seguinte forma: na Seção 1, o tema é introduzido e é apresentada a motivação para o surgimento da tecnologia OPC, elencando-se os problemas encontrados na indústria para a comunicação entre os diversos dispositivos e sistemas existentes no chão de fábrica naquela época; na Seção 2, é apresentado o princípio de funcionamento da tecnologia OPC, abordando-se as principais características do

padrão como, por exemplo, a estrutura cliente/servidor; na Seção 3, as variações mais relevantes do padrão original são apropriadamente explicadas, sendo elas: *Data Access*, *Historical Data Access*, *Alarms & Events* e *Unified Architecture*, além de um tópico descrevendo a motivação para o surgimento desta última; na Seção 4, são apresentados alguns exemplos de aplicação da tecnologia OPC e, em conclusão, na Seção 5, são apresentadas as considerações finais sobre a importância e a evolução do padrão OPC, além de considerações sobre o cumprimento dos objetivos do artigo e perspectivas para trabalhos futuros.

2 CONSIDERAÇÕES FINAIS

O artigo desenvolvido como objetivo do presente trabalho de conclusão de curso apresentou um levantamento sobre os conceitos fundamentais das tecnologias desenvolvidas pela OPC Foundation, desde meados da década de 1990 até a primeira década dos anos 2000. Foram abordados os problemas existentes na área de controle e aquisição de dados na época do surgimento da tecnologia, as soluções apresentadas a partir da publicação da primeira versão do OPC e a evolução representada pelas várias especificações publicadas pela OPC Foundation, até o OPC *Unified Architecture*.

Os princípios de funcionamento da tecnologia OPC foram apresentados de maneira generalizada, como uma introdução ao tema e, posteriormente, de maneira precisa para cada especificação OPC, como uma forma de mostrar as semelhanças e diferenças entre as diversas versões da tecnologia e fornecer ao leitor um texto orientado à evolução temporal das tecnologias da publicadas pela OPC Foundation.

A modalidade escolhida para o Trabalho de Conclusão de Curso – artigo de revisão bibliográfica – permitiu a apresentação de uma bibliografia introdutória, escrita em língua portuguesa, a cerca de um tema muito relevante para profissionais envolvidos com automação industrial. Em especial, o texto desenvolvido também pode ser usado como referência para os estudantes que cursam a disciplina Informática Industrial (8º período do curso de bacharelado em Engenharia de Controle e Automação da Universidade Federal de Lavras) que, em seu conteúdo programático, prevê aulas sobre os conceitos fundamentais da tecnologia OPC.

Dessa forma, o texto desenvolvido buscou apresentar o conteúdo de forma concisa, contendo as informações necessárias para que o leitor venha a ter conhecimento sobre os conceitos fundamentais da tecnologia OPC, suas variações e aplicações. A profundidade do texto foi planejada para estar condizente com o tempo de leitura recomendado para estudantes de graduação matriculados em uma disciplina de seis créditos, sendo que o assunto tratado deve ocupar uma das dezessete semanas do curso.

REFERÊNCIAS

Coelho, M.T.; Coelho Junior, E.M.; Quintino, L.F.; Piazza, C.A.D; Andrade, A.A. **A evolução das tecnologias OPC como subsídio para as fábricas inteligentes**. In: CEEL - Conferência de Estudos em Engenharia Elétrica, 14, 2016, Uberlândia.

Mahnke, W.; Leitner, S.; Damm, M. **OPC Unified Architecture**. Berlin: Springer, 2009.

OPC Foundation. **OPC Unified Architecture: Interoperability for Industrie 4.0 and the Internet of Things**. 2019.

Schwarz, M.H.; Börcsök, J. **A Survey on OPC and OPC-UA: About the standard, developments and investigations**. in: Internacional Conference on Information, Communication and Automation Technologies (ICAT), 24, 2013, Sarajevo.

Visaya. **OPC and OPC UA: Windows and industrial devices**. Disponível em: <<https://visaya.solutions/en/article/opc-and-opc-ua/>>. Acesso em 4 de fevereiro de 2020.

SEGUNDA PARTE

Uma revisão sobre a tecnologia OPC, suas variações e aplicações

Geovanne V. Figueiredo*.

**Universidade Federal de Lavras, Lavras, MG - Brasil
(e-mail: vichiattoge@gmail.com)*

Abstract: OPC (OLE for Process Control, originally; Open Platform Communications, nowadays) is a communication and interoperability standard for devices and systems in industrial networks. Such technology was developed in contrast to the large number of protocols and bus systems existing in the industry in the 1990s. During its consolidation, the OPC Foundation was created to manage the standard and several revisions and variations of the original standard were published, allowing, among other things, access to data stored in Programmable Logic Controllers (PLCs), historical data and the triggering of alarm and event notifications generated by supervisory systems. In addition, a new version of the technology was subsequently published, unifying the interfaces of all previous specifications and ensuring independence from third-party products. Due to the relevance of this subject, this article aims to present the fundamental concepts about OPC technology.

Resumo: O OPC (OLE *for Process Control*, quando surgiu; *Open Platform Communications*, na atualidade) é um padrão de comunicação e interoperabilidade para dispositivos e sistemas em redes industriais. Tal tecnologia foi desenvolvida em contraponto ao grande número de protocolos e meios de transporte existentes na indústria na década de 1990. Durante sua consolidação, a OPC Foundation foi criada para gerenciá-lo e esta publicou diversas revisões e variações do padrão original, permitindo, entre outras coisas, o acesso a dados armazenados em Controladores Lógico Programáveis (CLPs), dados em histórico e o disparo de notificações de alarmes e eventos gerados por sistemas de supervisão. Além disso, posteriormente, foi publicada uma nova versão da tecnologia, unificando as interfaces de todas as especificações anteriores e garantindo independência com relação a produtos de terceiros. Devido à relevância deste assunto, o presente artigo tem como objetivo apresentar os conceitos fundamentais sobre a tecnologia OPC.

Keywords: OPC Foundation; Interoperability; Communication standard; Data access; Classic OPC; OPC UA

Palavras-chaves: OPC Foundation; Interoperabilidade; Padrão de comunicação; Acesso a dados; OPC Clássico; OPC UA.

1. INTRODUÇÃO

Desde meados da década de 1980, a crescente complexidade dos processos industriais e o crescente número de dispositivos presentes em uma planta levaram pesquisadores e engenheiros a estudar e estabelecer padrões de redes, protocolos e meios de transporte de dados para a utilização em sistemas de automação (Schwarz e Börcsök, 2013).

Diferentes empresas desenvolveram, ao longo dos anos, diversas soluções para a comunicação entre os dispositivos, levando ao surgimento de mais de 50 protocolos de rede industrial e sistemas para transporte de dados (Schwarz e Börcsök, 2013). A quantidade de protocolos existentes, conectando um grande número de dispositivos dos mais variados modelos, de diferentes fornecedores, elevou a complexidade para se estabelecer conexões confiáveis entre os dispositivos, sendo que um padrão de interoperabilidade passou a ser necessário (Coelho et al., 2016).

Nos sistemas industriais, pode haver um grande número de dispositivos de diversos fabricantes, como sistemas SCADA (sigla em inglês para *Supervisory Control and Data Acquisition*), PEOs (Painéis Eletrônicos de Operação) além de aplicações instaladas em PCs (*Personal Computers*) convencionais, como as baseadas no SO (Sistema Operacional) Windows. Em geral, esses diversos dispositivos devem coletar e enviar dados aos CLPs (Controladores Lógico Programáveis), que por sua vez são responsáveis por comandar os atuadores, de acordo com sua programação, a qual muitas vezes se baseia nos dados que recebem dos sensores da planta (Schwarz e Börcsök, 2013).

No cenário anterior ao surgimento da tecnologia OPC (OLE *for Process Control*), os fabricantes precisavam fornecer *drivers* de comunicação para os sistemas SCADA e PEOs de forma específica (um para cada família de CLP), tarefa não trivial e propensa a diversas falhas (Schwarz e Börcsök, 2013). Um exemplo dessa situação pode ser observado na Figura 1.

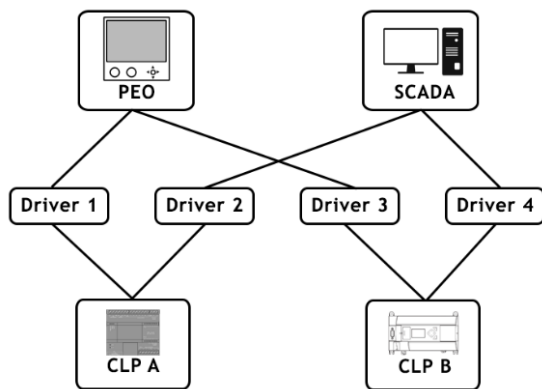


Figura 1 - Comunicação entre dispositivos utilizando *drivers* específicos (um para cada família de CLP), no cenário anterior ao surgimento da tecnologia OPC.

Com a utilização de diversos *drivers* para comunicação entre dispositivos, vários problemas podiam ocorrer, entre eles:

- **Versões simplificadas:** nem todas as funcionalidades dos diversos tipos de *hardware* eram suportadas por determinados *drivers*;
- **Retrabalho:** cada fornecedor de equipamentos precisava desenvolver diferentes *drivers* para comunicação com cada família de CLP;
- **Dificuldade de gerenciamento de versões:** atualizações de *firmware* dos CLPs poderiam levar ao mal funcionamento de alguns *drivers*, enquanto atualizações nas especificações de protocolos, caso fossem adotadas em um determinado processo, poderiam acarretar em prolongados tempos de manutenção;
- **Conflito no acesso a dispositivos:** em situações como a descrita na Figura 1, na qual diferentes implementações (*Driver 1* e *Driver 2*) acessam um mesmo dispositivo (CLP A), poderia haver perda de desempenho ou mesmo o não estabelecimento da comunicação, visto que imperfeições em um dos *drivers* poderiam exigir muito ou dominar os recursos de comunicação do CLP (OPC Foundation, 1998; Schwarz e Börcsök, 2013; Shimanuki, 1999).

Dessa forma, devido aos problemas supracitados, grande parte dos esforços das empresas quando do projeto e manutenção de seus sistemas SCADA e PEOs era gasta com a definição *drivers* de comunicação (Filho, 2005). Paralelamente a este cenário, na década de 1990, a Microsoft lançou a família 3.x do SO Windows, introduzindo soluções para a execução simultânea de aplicações. Além disso, a Microsoft também introduziu a tecnologia DDE (*Dynamic Data Exchange*), permitindo a troca de dados, em tempo real, entre essas aplicações (OPC Connect, 2020).

Posteriormente, em 1992, a Microsoft introduziu a tecnologia OLE (*Object Linking and Embedding*) 2.0, que é uma coleção de objetos que fornece serviços para vincular e incorporar objetos de várias aplicações em funcionamento no SO Windows. A nova versão do OLE foi criada em conformidade com a tecnologia COM (*Component Object Model*), que é uma especificação para criação de objetos e exposição de interfaces

dos mesmos de maneira independente com relação à linguagens de programação. Desse modo, utilizando a tecnologia COM, uma aplicação (cliente) pode invocar métodos de objetos expostos em interface de outra aplicação (servidor), em execução no SO Windows. Por ter sido criada em conformidade com a tecnologia COM, o OLE também estava em conformidade com a tecnologia DCOM (*Distributed COM*), que é uma extensão à COM, permitindo o acesso a objetos de forma distribuída (em aplicações que estão sendo executadas em PCs separados) (Chung et. al, 1998; Botton, 1999).

A tecnologia OLE se mostrou mais flexível, robusta e com mecanismos de transporte mais eficientes do que o DDE. Desse modo, um grupo de estudiosos do ramo de controle e aquisição de dados começou a se reunir com o objetivo de desenvolver soluções baseadas nessa tecnologia para transportar dados de processos em tempo real (OPC Connect, 2020). O grupo era composto por membros das empresas Fisher-Rosemount, Rockwell Software, Opto 22, Intellution e Intuitive Technology e iniciou os trabalhos no ano de 1995. Já em dezembro daquele ano, o grupo publicou um rascunho de especificação para o padrão OPC, sendo que, em agosto de 1996, foi publicada a primeira versão oficial do padrão. Como o padrão foi desenvolvido baseado na tecnologia OLE e, consequentemente, baseado na tecnologia COM/DCOM, pode-se dizer que o OPC também se baseia na arquitetura cliente-servidor (OPC Foundation, 2019; Visaya, 2020).

Para desenvolver a nova especificação, a força tarefa OPC utilizou tecnologias já existentes da Microsoft, como o OLE, economizando tempo e esforços na especificação de protocolos e interfaces (Mahnke, Leitner e Damm, 2009; Schwarz e Börcsök, 2013). Como a utilização de aplicações e sistemas SCADA baseados no SO Windows era uma prática comum no ramo de controle e automação de processos desde aquela época, a tecnologia OPC foi rapidamente adotada na indústria. Dessa forma, o OPC tornou-se, na prática, um novo padrão para comunicação e interoperabilidade entre sistemas e CLPs (Mahnke, Leitner e Damm, 2009). Com isso, a força tarefa tomou a decisão de que a especificação do OPC deveria continuar sendo desenvolvida e que precisava ser gerenciada por uma organização independente e sem fins lucrativos. Essa organização foi formada e denominada “OPC Foundation” (OPC Connect, 2020).

A primeira versão do OPC (OPC 1.0) era simples, pois tinha apenas o propósito de abstrair os protocolos de CLPs em uma interface única, permitindo que sistemas SCADA e PEOs se comunicassem com um “intermediário” que, por sua vez, deveria converter requisições de leitura/escrita genéricas em requisições compreendidas pelos CLPs (OPC Foundation, 2020a). Na Figura 2 pode ser observado um exemplo esquemático do princípio de abstração apresentado pelo OPC 1.0.

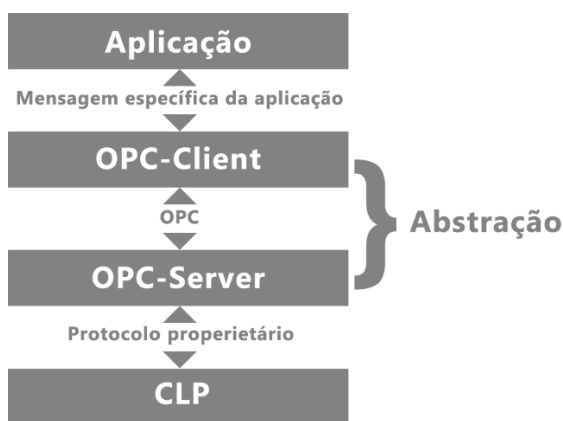


Figura 2 – Abstração dos protocolos proprietários por meio da tecnologia OPC.

Após o lançamento da primeira versão, por mais de uma década, o padrão OPC foi utilizado na indústria de maneira restrita a aplicações baseadas em tecnologias presentes no SO Windows. Neste período, diversas especificações foram publicadas pela OPC Foundation, descrevendo o funcionamento de interfaces entre aplicações de forma a permitir o acesso padronizado a dados de dispositivos de campo em tempo real (*OPC Data Access*), o acesso a dados armazenados em histórico (*OPC Historical Data Access*) e à notificações de alarmes e eventos (*OPC Alarms and Events*) (OPC Foundation, 2020a; Schwarz e Börcsök, 2013).

Com o avanço de sistemas com arquiteturas orientadas a serviços, a OPC Foundation desenvolveu uma nova versão de sua tecnologia – o OPC UA (*OPC Unified Architecture*). A versão 1.0 deste padrão foi publicada em 2006 em cooperação com a IEC (*Internacional Electrotechnical Commission*), tornando-se um padrão internacional de comunicação. Uma das principais diferenças apresentadas com relação às especificações anteriores foi a independência de plataformas externas, como o Windows. Além disso, possuía grande capacidade de modelagem de dados, permitindo a integração de sistemas presentes nas várias camadas da pirâmide da automação e escalável para os mais diferentes tipos de propósitos, como a utilização do conceito de “Internet das Coisas” e aplicações em Nuvem (OPC Foundation, 2020a).

No presente trabalho são apresentadas as características básicas e conceitos fundamentais das principais variações da tecnologia OPC. Na Seção 2, é apresentado o princípio de funcionamento da tecnologia OPC, abordando-se as principais características do padrão; na Seção 3, são apresentadas as variações mais relevantes do padrão original: *OPC Data Access*, *OPC Historical Data Access*, *OPC Alarms & Events* e *OPC Unified Architecture*; na Seção 4, são apresentados alguns exemplos de aplicação da tecnologia OPC e na Seção 5, são apresentadas as considerações finais sobre a importância e a evolução do padrão OPC.

2. PRINCÍPIO DE FUNCIONAMENTO

2.1 Conceitos Fundamentais

Conceitualmente, o OPC pode ser compreendido como um elemento de abstração entre os dispositivos de campo e os

dispositivos e pacotes de *software* voltados à supervisão e controle de processos. Dessa forma, estes últimos podem solicitar e enviar dados sem que seja necessário conhecer qualquer informação muito específica sobre os primeiros (Kominek, 2009; Youngsheng, Miaomiao e Fengzhi, 2013).

A tecnologia OPC foi construída com base na arquitetura cliente-servidor, em que o cliente é o elemento ativo da comunicação (Junior et al., 2016; Kominek, 2009). Logo, a abstração dos dispositivos é alcançada por meio de dois componentes específicos: OPC-Server e OPC-Client (Kominek, 2009). Uma aplicação requer acesso a um dispositivo comunicando-se, por meio do seu OPC-Client, com um OPC-Server, que é responsável pela comunicação direta com o dispositivo de campo (Junior et al., 2016; Postól e Zbrzezny, 2006). Desse modo, as aplicações-cliente sempre tomam iniciativa na troca de dados, enquanto os servidores apenas respondem às requisições (Kominek, 2009).

Assim, um coletor de dados (por exemplo, um supervisor) faz uma requisição ao seu OPC-Client. Este último se comunica com o OPC-Server que, por sua vez, se comunica com uma fonte de dados (por exemplo, um CLP). Em seguida, a resposta da requisição percorre o mesmo caminho, mas no sentido contrário, levando as informações da fonte até o coletor (Junior et al., 2016; Kominek, 2009).

Tanto a comunicação entre o coletor de dados e o OPC-Client quanto a comunicação entre o OPC-Server e a fonte de dados são realizadas utilizando-se protocolos nativos dos respectivos dispositivos (Kominek, 2009). Já a comunicação entre o cliente e o servidor é definida por meio da tecnologia COM/DCOM (Postól e Zbrzezny, 2006). Essa comunicação pode ocorrer por meio da troca de dados entre aplicações em execução em um mesmo PC (utilizando a especificação COM) ou, com a ajuda de algum protocolo de rede, entre aplicações executadas em PCs distintos (utilizando a especificação DCOM) (Thompson et. al, 1997). Um exemplo, em diagrama esquemático, da comunicação entre dispositivos e sistemas utilizando o OPC pode ser observado na Figura 3.

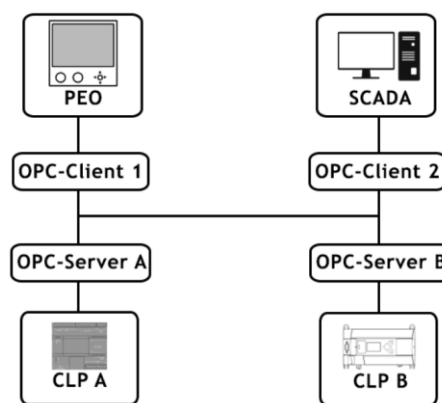


Figura 3 – Diagrama esquemático de uma comunicação entre dispositivos utilizando a tecnologia OPC.

2.2 Servidores OPC

O OPC-Server é um dos elementos fundamentais do OPC e é uma implementação em *software* que inclui pelo menos um trecho de código específico para comunicação com um

dispositivo de campo ou base de dados (Kominek, 2009; OPC Foundation, 1998).

Em seu funcionamento, o OPC-Server pode ser interpretado como um “tradutor”, que converte as requisições recebidas no padrão OPC para o protocolo nativo de um dispositivo de campo e vice-versa, funcionando de maneira bidirecional, já que pode ler e escrever informações no dispositivo (Kominek, 2009).

Conceitualmente, o OPC-Server pode ser separado em três partes: o módulo de comunicação OPC, o módulo de tradução e o módulo de comunicação nativa (Kominek, 2009). Na Figura 4, pode-se observar a divisão e o relacionamento dessas três partes.



Figura 4 - Estrutura conceitual de um OPC-Server.

O módulo de comunicação OPC é responsável por realizar efetivamente a comunicação entre o servidor e o cliente (Kominek, 2009). Para a efetivação da comunicação, o servidor expõe diversos objetos em sua interface e o cliente realiza chamadas de métodos nestes, realizando buscas por dados e requisições de leitura/escrita no servidor (OPC Foundation, 1998).

Por sua vez, o módulo de tradução tem a responsabilidade de interpretar as requisições vindas do cliente e transformá-las em requisições compreensíveis para o dispositivo conectado ao servidor. Analogamente, este módulo também deve tratar os dados lidos do dispositivo para que sejam compreendidos pelo cliente (Kominek, 2009).

Por fim, como se espera, o módulo de comunicação nativa faz referência à interface de comunicação entre o servidor e o dispositivo de campo. Tal comunicação se dá por meio do protocolo nativo do dispositivo e isso faz com que o servidor atue como um *driver* padrão para o dispositivo em questão. Dessa forma, o dispositivo não precisa ser previamente concebido para comunicar-se com um servidor OPC que, por sua vez, deve apenas ser capaz de implementar o método nativo de comunicação do primeiro para que consiga abstraí-lo em uma interface padrão exposta para clientes OPC (Kominek, 2009).

2.3 Clientes OPC

Os OPC-Clients são pacotes de *software* desenvolvidos para comunicar-se com aplicações e OPC-Servers, atendendo as especificações da OPC Foundation. A implementação dos clientes pode se dar na forma de um módulo embarcado em

outra aplicação ou na forma de um módulo individual (Junior et al., 2016; Kominek, 2009).

O funcionamento interno do OPC-Client é análogo ao funcionamento do seu par, o OPC-Server. O cliente traduz uma requisição da aplicação em que está embarcado em uma requisição compreendida pelo servidor, que retorna ao primeiro a informação requisitada. Conseqüentemente, o cliente também precisa traduzir essa última informação para um modelo compreendido pela aplicação (Kominek, 2009).

Assim como o OPC-Server, o OPC-Client também pode ser separado em três elementos conceituais: o módulo de comunicação com a aplicação, o módulo de tradução e o módulo de comunicação OPC (Kominek, 2009). A Figura 5 mostra a divisão e o relacionamento das três partes desse elemento.



Figura 5 – Estrutura Conceitual de um OPC-Client.

O módulo de comunicação com a aplicação é análogo ao módulo de comunicação nativa do OPC-Server. Este módulo é responsável por executar a comunicação com a aplicação em que o cliente está embarcado ou está conectado (caso o cliente esteja implementado na forma de um módulo individual). Em ambos os casos, a comunicação se dá por meio do protocolo específico da aplicação. Na hipótese de a aplicação consistir em, por exemplo, um PEO que não inclua um cliente OPC embarcado, a comunicação se dá por meio do protocolo nativo do mesmo. Por sua vez, caso a aplicação seja executada em um PC, se o cliente consiste em um módulo embarcado, intrinsecamente a aplicação trocará dados com o cliente por meio de mensagens internas; por outro lado, se o cliente consiste em um módulo individual, a comunicação se dá por meio da troca de mensagens por algum meio disponibilizado pelo fornecedor da aplicação (Kominek, 2009).

Por sua vez, o módulo de tradução recebe as requisições da aplicação e as traduz para o formato de dados OPC, tornando possível ao servidor interpretar as requisições e executá-las. Como deve funcionar de maneira bidirecional, o módulo de tradução também deve converter dados vindos do servidor para o formato compreendido pela aplicação (Kominek, 2009).

Por fim, o módulo de comunicação OPC é o responsável por realizar efetivamente a comunicação com seu par, o OPC-Server. A comunicação é efetivada por meio da utilização da tecnologia COM/DCOM (Kominek, 2009; OPC Foundation, 1998). O cliente acessa a interface padrão exposta pelo servidor e, dessa forma, é capaz de invocar métodos de objetos e trocar dados com o mesmo (Mahnke, Leitner e Damm, 2009; OPC Foundation, 1998; Son e Yi, 2010).

3. TECNOLOGIAS OPC

3.1 – Especificações OPC

A primeira especificação do OPC foi lançada em 29 de agosto de 1996, com o nome “OPC Specification”, descrevendo uma tecnologia de troca de dados entre dispositivos utilizando tecnologias já existentes da Microsoft (Mahnke, Leitner e Damm, 2009; OPC Connect, 2020). Desde então, várias outras especificações foram lançadas pela OPC Foundation, detalhando diversas interfaces e diferentes formas de acesso a dados distintos (Visaya, 2020).

Em 1997, foi lançada uma correção para especificação OPC 1.0, chamada de OPC 1.0A, agora conhecida como OPC *Data Access*, ou simplesmente OPC DA (OPC Connect, 2020). Posteriormente, em complemento, a OPC Foundation lançou outras especificações, com destaque para OPC A&E (Alarms and Events) e OPC HDA (*Historical Data Access*), sendo que, atualmente, o conjunto formado por essas três tecnologias é conhecido como Classic OPC (Mahnke, Leitner e Damm, 2009).

Por sua grande importância no mercado de automação industrial, chegando a ser implementada em 99% dos produtos que utilizam a tecnologia OPC (Mahnke, Leitner e Damm, 2009), a especificação OPC DA recebeu mais atenção por parte da OPC Foundation, chegando à versão 3.0 (OPC Foundation, 2003a). Por sua vez, a especificação OPC A&E foi atualizada somente até a versão 1.1 (OPC Foundation, 2002), enquanto a OPC HDA foi atualizada até a versão 1.2 (OPC Foundation, 2003b).

Além dessas, a OPC Foundation ainda lançou as especificações OPC *Overview* e OPC *Commom*, definindo interfaces e comportamentos comuns a todas as especificações baseadas na tecnologia COM/DCOM (Mahnke, Leitner e Damm, 2009). Foram publicadas ainda, as especificações OPC *Complex Data*, OPC *Batch* e OPC *DX (Data eXchange)*, como extensões à OPC DA, além da XML-DA (*XLM Data Access*). Esta última (considerada precursora da tecnologia OPA UA, que será abordada mais a frente) baseia-se na especificação OPC DA, porém utiliza *web services* em detrimento da tecnologia COM/DCOM (Schwarz e Börcsök, 2013). Por fim, foi publicada a especificação OPC *Security*, bem como iniciado o desenvolvimento da especificação OPC *Commands*, que nunca chegou a ser lançada (Mahnke, Leitner e Damm, 2009). A Figura 6 ilustra as especificações publicadas nesta fase (anterior à especificação OPC UA).

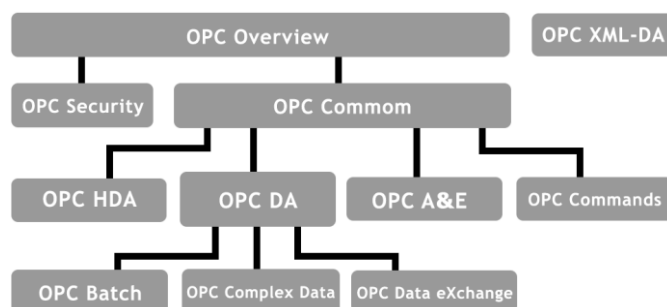


Figura 6 - Especificações OPC antes da tecnologia OPC UA.

Posteriormente, representando um novo marco da tecnologia OPC, a especificação *Unified Architecture* foi desenvolvida para oferecer um padrão aberto de comunicação sem nenhuma dependência ou ligação com protocolos e tecnologias proprietárias de fornecedores individuais. O OPC UA pode ser considerado a segunda geração da tecnologia OPC, estendendo significativamente seu alcance (OPC Foundation, 2019).

3.2 OPC Data Access

Como visto, a especificação OPC DA nasceu de uma correção feita na primeira especificação do OPC, sendo a mais utilizada dentre das especificações conhecidas como Classic OPC pois tornou-se, na prática, um padrão de comunicação em redes industriais (Mahnke, Leitner e Damm, 2009; OPC Connect, 2020; Schwarz e Börcsök, 2013).

O OPC DA permite que uma aplicação, por meio de seu cliente, envie requisições de leitura e escrita de dados, por meio de um servidor, a um destinatário (OPC Foundation, 1999). Desse modo, pode-se dizer que a principal função do OPC DA é promover a troca de dados entre esta aplicação (que, entre outros, pode ser um sistema SCADA ou um PEO) e este destinatário (que, na maioria dos casos, é um CLP) (Mahnke, Leitner e Damm, 2009).

Para que seja possível mover dados entre o servidor e o cliente OPC DA, uma conexão é estabelecida entre eles. Para isso, são utilizados os objetos disponibilizados na interface do servidor (também conhecidos como objetos COM) que, organizados hierarquicamente do topo para a base, são: OPCServer, OPCGroup e OPCItem (OPC Foundation, 1998). Para estabelecer uma conexão, o cliente cria um objeto do tipo OPCServer, que contém informações do servidor e funciona como um contêiner para o objeto OPCGroup, sendo possível que um objeto OPCServer contenha vários objetos OPCGroup (Liu, Cai e Peng, 2009; OPC Foundation, 1998). O objeto OPCServer oferece ainda métodos para que o cliente navegue pelos objetos na hierarquia do servidor para buscar itens e propriedades como tipos de dados e direitos de acesso (OPC Foundation, 2001).

Contido em um objeto OPCServer, o objeto OPCGroup fornece uma maneira para que o cliente organize os dados, sendo ainda um contêiner de objetos do tipo OPCItem, podendo conter vários deles (Liu, Cai e Peng, 2009). A organização dos objetos do tipo OPCItem é realizada pelo OPCGroup por meio da definição de características de leitura e atualização dos mesmos, agrupando-se itens com configurações idênticas (OPC Foundation, 1998; Son e Yi, 2010). A definição das características de leitura e atualização prevê os seguintes comportamentos:

- Tipo de leitura:
 - Leitura síncrona: o servidor aguarda a confirmação de recebimento por parte do cliente antes de enviar novos dados;
 - Leitura assíncrona: o servidor não precisa aguardar nenhuma confirmação por parte do cliente antes de enviar novos dados.
- Atualização dos dados:

- Leitura cíclica: uma vez definida a taxa de atualização, que representa o tempo de ciclo da comunicação, o cliente faz requisições periódicas ao servidor;
- Leitura por mudança de estado: o servidor envia ao cliente os itens que sofreram uma mudança em seu valor acima de uma banda morta pré-configurada (ou sofreram mudança no estado de qualidade). Nesse contexto, a banda morta é utilizada para reduzir a quantidade de dados que o servidor envia ao cliente, desprezando mudanças que possam ser consideradas irrelevantes (Son e Yi, 2010).

Por outro lado, os comandos de escrita enviados pelo cliente são prontamente executados pelo servidor, podendo ou não haver solicitação de confirmação por parte do primeiro (OPC Foundation, 2001).

Por fim, o objeto OPCItem representa uma conexão com o dispositivo de campo, contendo em si três atributos: *value*, *timestamp* e *quality* (Liu, Cai e Peng, 2009; OPC Foundation, 1998). O atributo *value* é o valor de um dado lido pelo servidor (Schwarz e Börcsök, 2013). O atributo *timestamp* informa o momento em que a informação contida no atributo *value* foi obtida. O atributo *quality* representa o estado do *value*, podendo ser “Bad”, quando o valor não é utilizável, “Uncertain” quando a qualidade do valor é incerta ou “Good”, quando não há nenhum problema com a leitura do valor (OPC Foundation, 2003a).

3.3 OPC Historical Data Access

A especificação *Historical Data Access* pode ser considerada uma extensão da especificação *Data Access*, porém, em vez de fornecer acesso a dados produzidos em tempo real, o OPC HDA fornece um meio de acesso a dados armazenados em sistemas historiadores (Mahnke, Leitner e Damm, 2009; Schwarz e Börcsök, 2013). Dessa forma, uma aplicação com um cliente HDA embarcado pode ler, inserir, alterar e excluir dados de um sistema historiador por meio do seu servidor HDA (OPC Foundation, 2003b).

O funcionamento da especificação OPC HDA é similar ao da OPC DA. Para conectar-se a um servidor, o cliente cria o objeto OPCHDAServer neste primeiro (Mahnke, Leitner e Damm, 2009), que passa a fornecer os métodos necessários para a manipulação de dados no sistema historiador. Também é definido um segundo objeto, o OPCHDABrowser, que permite ao cliente navegar no espaço de endereços do servidor (OPC Foundation, 2003b).

O OPC HDA fornece três métodos para a leitura dos dados armazenados: no primeiro, o cliente HDA define as variáveis que serão acessadas, a quantidade de valores de cada uma e o período de tempo, sendo que o servidor retorna os dados arquivados que correspondem à essa requisição; com o segundo método, o cliente define as variáveis que quer ter acesso com *timestamps* específicos, em vez de um intervalo de tempo; no terceiro, o cliente define as variáveis e um período de tempo para ler valores “agregados” dos arquivos, isto é, valores como média, valor máximo ou mínimo, primeiro ou último valor do intervalo, amplitude, variância, desvio padrão,

entre outros (Mahnke, Leitner e Damm, 2009; OPC Foundation, 2003b). As operações para agregar dados são realizadas no servidor e depois enviadas ao cliente (OPC Foundation, 2003b).

Assim como no OPC DA, a especificação *Historic Data Access* fornece atributos *quality* e *timestamp* para os dados lidos (Mahnke, Leitner e Damm, 2009). O atributo *quality* influencia o método de cálculo de dados agregados por parte do servidor: valores que apresentam atributo de qualidade “Bad” são omitidos no cálculo de agregados, e o resultado do cálculo é associado a um atributo *quality* como “Uncertain” ou “Subnormal” (OPC Foundation, 2003b).

3.4 OPC Alarms & Events

A especificação *OPC Alarms & Events* é destinada à sistemas que geram alarmes e eventos em um processo (Mahnke, Leitner e Damm, 2009). Conforme definido no escopo desta especificação, um alarme é uma condição anormal, sendo tratado como um caso especial de um estado presente em um servidor que seja de interesse de um cliente. Por outro lado, um evento é uma ocorrência detectável que é relevante para o servidor, para o dispositivo que este representa ou para o cliente (OPC Foundation, 2002).

A especificação OPC A&E complementa as especificações DA e HDA, mas também pode ser implementada de forma independente (OPC Foundation, 2002). Seu funcionamento assemelha-se ao dessas últimas, sendo que um cliente se conecta ao servidor criando o objeto OPCEventServer. A diferença é que, quando isso acontece, o objeto OPCEventSubscription também é gerado, que por sua vez pode ser entendido como uma “assinatura” de notificações de alarmes e eventos (Mahnke, Leitner e Damm, 2009). Outra diferença com relação à especificação OPC DA é que o OPC A&E não implementa requisições para informações específicas, de modo que, uma vez “inscrito” no servidor, o cliente receberá as notificações sempre que estas ocorrerem. Para diminuir a quantidade de notificações recebidas, o cliente pode definir critérios de filtragem no servidor (Mahnke, Leitner e Damm, 2009).

As notificações de alarmes e eventos acontecem por meio das condições, que são representadas por objetos OPCCCondition. Os objetos OPCCCondition são associados a objetos OPCSource, que por sua vez podem fazer referência ao valor de uma variável, um dispositivo ou um sistema. Caso o servidor A&E também seja um servidor DA, o objeto OPCSource pode ser um OPCItem (OPC Foundation, 2002). O objeto OPCCCondition apresenta vários atributos para denotar uma condição, entre eles: *name*, que é único para cada condição, *quality*, *acked*, que denota se a notificação foi recebida por um cliente e *acknowledgerID*, que é o ID do Client que a recebeu (OPC Foundation, 2002).

3.5 OPC Unified Architecture – Motivação

As especificações que formam o grupo conhecido como Classic OPC foram projetadas para operar como interfaces para *drivers* de comunicação, possibilitando que elementos

como sistemas SCADA e PEOs pudessem realizar operações de leitura e escrita na memória de dispositivos como os CLPs de forma padronizada (Mahnke, Leitner e Damm, 2009). Porém, sua utilização era impossível fora de plataformas baseadas no SO Windows, por conta da dependência da tecnologia COM/DCOM (Son e Yi, 2010). Além disso, por vezes, a tecnologia DCOM se mostrava ineficiente para o acesso remoto a dados (Mahnke, Leitner e Damm, 2009).

Outras dificuldades foram encontradas para a aplicação em maior escala da tecnologia OPC, principalmente com relação ao uso de várias especificações simultaneamente – vários clientes e servidores eram necessários já que não era possível utilizar interativamente elementos de especificações diferentes (Schwarz e Börcsök, 2013). A Figura 7 ilustra esse cenário, em que clientes não conseguem obter dados de servidores de outras especificações.

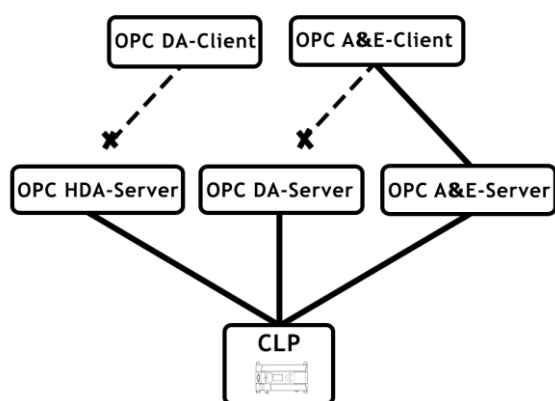


Figura 7 – Incapacidade de comunicação entre servidores e clientes OPC de especificações distintas.

Com o objetivo de proporcionar uma solução para comunicação de forma segura, robusta, com alta escalabilidade e interoperabilidade, com possibilidade de redundância de informações, alta performance em redes locais e acesso externo, por meio de *firewalls* e, principalmente, independente de fornecedores e de plataformas (OPC Foundation, 2019), a OPC Foundation desenvolveu e testou durante a primeira década dos anos 2000 uma nova especificação, publicando em 2007 o *OPC Unified Architecture*, em substituição às especificações clássicas (OPC Foundation, 2017a; Steinkrauss, 2010).

Para facilitar a implementação do OPC UA na base de produtos que utilizavam as especificações anteriores, a especificação foi desenvolvida para ser compatível com elas (OPC Foundation, 2017a; Son e Yi, 2010). Além disso, para facilitar a sua adoção a nível mundial, o OPC UA foi desenvolvido em conformidade com os requisitos da IEC. Desse modo, o OPC UA (IEC62541) pode ser considerado um padrão internacional de comunicação (OPC Foundation, 2019).

Com a introdução do OPC UA, a OPC Foundation passou a tratar o OPC Classic como uma versão anterior de sua tecnologia. Além disso, com o abandono da tecnologia OLE, da Microsoft, a designação *OLE for Process Control* (que formava o acrônimo OPC) perdeu o sentido e foi substituída pelo termo *Open Platform Communications*, propositalmente

escolhido para não alterar a sigla pela qual a tecnologia tornou-se vastamente conhecida (OPC Foundation, 2020a).

3.6 OPC Unified Architecture – Princípios de Funcionamento

Assim como as especificações que formam o OPC Classic, o OPC UA utiliza a arquitetura cliente-servidor, porém com sua arquitetura baseada no paradigma SOA (orientação a serviços, do termo em inglês *Service-Oriented Architecture*) (OPC Foundation, 2019; Steinkrauss, 2010). Em geral, o OPC UA é uma infraestrutura de comunicação baseada em camadas, sendo que seus componentes fundamentais são o mecanismo de transporte e a modelagem de dados (Mahnke, Leitner e Damm, 2009).

O mecanismo de transporte do OPC UA define diferentes métodos de transporte de informação para diferentes objetivos, baseando-se em uma “pilha de comunicação” formada pelos seguintes componentes: codificação de dados, a segurança da comunicação e o transporte de dados. Para que a codificação e o transporte de dados pudessem ocorrer de forma eficiente, o grupo de trabalho do OPC UA desenvolveu uma tecnologia baseada no protocolo TCP, conhecida como TCP UA binário. Porém, outros protocolos podem ser usados em conformidade com o OPC UA, permitindo troca de dados utilizando *web services* ou HTTPS, entre outras tecnologias (Mahnke, Leitner e Damm, 2009).

Por sua vez, a modelagem de dados pode ser considerada a maior inovação do OPC UA com relação ao OPC Classic (Mahnke, Leitner e Damm, 2009), definindo as regras e os blocos de construção básicos necessários para expor informações (OPC Foundation, 2019). Os modelos de dados são baseados no paradigma de orientação a objetos e permitem a descrição de equipamentos, sistemas e funções variadas.

No OPC UA, o servidor trabalha com diferentes tipos de informações na sua interface, agrupando todos os tipos de dados presentes nas especificações OPC Classic. Dessa forma, um único servidor é capaz de mostrar dados em tempo real (como um OPC DA-Server) e em histórico (como um OPC HDA-Server), notificar alarmes e eventos (como um OPC A&E-Server), além de outros tipos de dados relacionados às outras especificações da geração anterior (OPC Foundation, 2019; OPC Foundation, 2017d). Para ser capaz de disponibilizar a um cliente tais informações, os objetos disponibilizados na interface do servidor são baseados em atributos, métodos e notificadoros de eventos, como mostrado na Figura 8. Por sua vez, um cliente pode definir maneiras apropriadas de mostrar as informações contidas nesses objetos, bem como utilizá-los para manipular dados de configuração (OPC Foundation, 2017b).

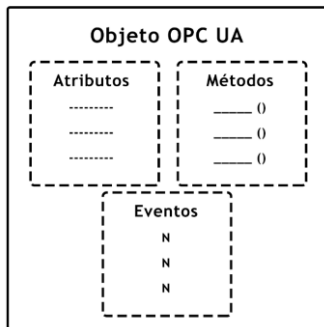


Figura 8 - Base de um objeto OPC UA.

A interface de um servidor é disponibilizada ao cliente por meio dos *UA Services*, que consistem em operações de chamada que o último faz no primeiro. Os *services* podem ser do tipo *request/response* ou *subscription*. No tipo *request/response*, um cliente requisita uma informação do servidor, que naturalmente a responde. No tipo *subscription*, um cliente se inscreve em um servidor para receber notificações periódicas, que podem ser: eventos, alarmes ou mudanças de dados em objetos (OPC Foundation, 2017c).

Como é definido de maneira independente de fornecedores, protocolos e plataformas específicas, o OPC UA pode ser implementado desde dispositivos baseados em 16-bits até computadores atuais com arquitetura baseada em 64-bits, em diversos sistemas operacionais (Steinkrauss, 2010). A sua independência com relação às linguagens de programação e a sistemas específicos torna o OPC UA um padrão de comunicação tão flexível que pode ser empregado em todos os níveis da pirâmide de automação, em qualquer ramo industrial (Steinkrauss, 2010).

4. CENÁRIO ATUAL DAS TECNOLOGIAS OPC

Após mais de duas décadas desde a sua formação, a OPC Foundation contava, em 2019, com mais de 730 membros (OPC Foundation, 2019). Além disso, no primeiro semestre de 2020, um levantamento indicou que as tecnologias OPC estavam presentes em mais de 17 milhões de máquinas, em fábricas das mais diversas partes do planeta (OPC Foundation, 2020b).

O OPC tem sido utilizado, por exemplo, por ferramentas como Excel (Junior et al., 2016) e aplicações simples da especificação OPC DA, como protocolo de acesso à memória de CLPs. Outro exemplo de utilização do OPC UA é o desenvolvimento de blocos de funções, permitindo aos CLPs a possibilidade de atuarem como clientes, chamando verticalmente métodos em sistemas MES/ERP (*Manufacturing Execution System/Enterprise Resource Planning*) ao mesmo tempo em que podem trocar dados horizontalmente com outros controladores na rede (OPC Foundation, 2019).

Mais especificamente, desde a publicação da especificação OPC UA, em 2007, pesquisas têm sido feitas com o intuito de avaliar a possibilidade de seu uso em conjunto com diversas tecnologias. Em seu trabalho, Merz (2019) cita que, considerando os conceitos básicos da Indústria 4.0, o OPC UA fornece as tecnologias necessárias para comunicação M2M (*Machine-to-Machine*) e IoT (*Internet of Things*); em outro

estudo, Durkop, Czybik e Jasperneite (2015) concluíram que o OPC UA apresentou melhor performance do que outros protocolos, como EDGE (*Enhanced Data rates for Global System for Mobile Evolution*), LTE (*Long Term Evolution*) e UMTS (*Universal Mobile Telecommunication System*) para comunicação M2M em uma rede de celular, em ambiente laboratorial.

Por fim, pode-se ressaltar que, a OPC Foundation disponibiliza em seu *website* (opcfoundation.org) sessões com notícias sobre utilização de sua tecnologia e estudos de caso em vários segmentos, como na indústria automotiva, química, alimentícia, petróleo e gás, além de monitoramento de consumo de energia, tratamento de água e transporte (OPC Foundation, 2020b).

5. CONCLUSÃO

O OPC evoluiu de uma tecnologia de abstração de protocolos proprietários para um padrão internacional de comunicação possibilitando a integração de sistemas e dispositivos em todas as camadas da pirâmide de automação. Tecnologias emergentes para troca de dados, como IIoT (*Industrial IoT*) e M2M encontram no OPC UA um padrão de modelagem de dados robusto, seguro e confiável.

A padronização de acesso a dados permitiu que PEOs e sistemas supervisórios se conectassem de maneira simples a CLPs. Atualmente, a interoperabilidade oferecida pelo padrão OPC UA permite que, em uma mesma fábrica, ocorram ao mesmo tempo as chamadas “integração horizontal” e “integração vertical” entre dispositivos, sendo que, neste último caso, incluem-se os sistemas em nuvem.

Com grande aceitação desde a publicação da primeira especificação, o OPC teve sua importância aumentada com a introdução de diversas tecnologias de comunicação na indústria.

O presente artigo foi escrito visando a apresentação dos conceitos fundamentais sobre a tecnologia OPC. Foi descrita a motivação para o surgimento do OPC, elencando-se os problemas existentes para a comunicação entre dispositivos e sistemas no chão de fábrica nos anos anteriores à publicação da primeira versão do padrão OPC. Além disso, foram descritos os princípios de funcionamento da tecnologia, abordando a estrutura de servidores e clientes. Por fim, descreveu-se as características das principais especificações publicadas pela OPC Foundation: *Data Access*, *Historical Data Access*, *Alarms & Events* e *Unified Architecture*, bem como uma breve descrição da motivação para o desenvolvimento desta última. Dessa forma, espera-se que o presente artigo venha a contribuir como uma bibliografia introdutória para pesquisadores, profissionais e estudantes envolvidos com o ramo da automação industrial.

Como perspectiva para trabalhos futuros, pode-se abordar o desenvolvimento mais recente da tecnologia, com pesquisa bibliográfica mais ampla sobre aplicações da especificação OPC *Unified Architecture*, tanto em meio industrial, quanto em meio acadêmico. Pode-se, ainda, destacar-se outros aspectos da supracitada especificação, como as outras camadas que a compõe.

6. REFERÊNCIAS

- Botton, D. (1999). Interfacing Ada 95 to Microsoft COM and DCOM technologies. In ACM SIGAda Ada Letters, 9-14.
- Chung, P.E., Huang, Y., Yajnik, S., Liang, D., Shih, J.C., Wang, C. and Wang, Y. (1998). DCOM and CORBA side by side, step by step, and layer by layer.
- Coelho, M.T., Junior, E.M.C, Quintino, L.F., Piazza, C.A.D and Andrade, A.A. (2016). A evolução das tecnologias OPC como subsídio para as fábricas inteligentes. In Conferencia de Estudos em Engenharia Elétrica.
- Durkop, L., Czybik, B. and Jasperneite, J. (2015). Performance evaluation of M2M protocols over cellular networks in a lab environment. In 18th International Conference on Intelligence in Next Generation Networks, 70-75.
- Filho, O.A.N. (2005). Desenvolvimento de servidores OPC DA e OPC XML DA para sistemas de aquisição de dados via telefonia celular. Universidade Federal do Espírito Santo.
- Junior, E.M.C, Filho, P.D., Quintino, L.F, Piazza, C.A.D, Coelho, M.T., Oliveira, W.B. and Andrade, A.A. (2016). Comunicação entre CLP - IHM - Excel por meio de protocolo OPC DA, estudo de caso. In IEEE/IAS International Conference on Industry Applications.
- Kominek, D. (2009). OPC: the ins and outs to what it's about: "the every man's guide to OPC".
- Liu, T., Cai, G. and Peng, X. (2009). OPC server software design in DCS. In International Conference on Computer Science & Education, 456-458.
- Mahnke, W., Leitner, S. and Damm, M. (2009). OPC unified architecture, 1st edn, Springer, Berlin.
- Merz, S. (2019). Intelligent water management - M2M interaction based on OPC UA. In OPC Unified Architecture: *interoperability for Industrie 4.0 and the Internet of Things*, OPC Foundation, version 10, 55.
- OPC Connect. (2020). History of OPC, viewed 31 January 2020, <<https://www.opcconnect.com/history.php>>.
- OPC Foundation. (1998). OPC overview: *version 1.0*.
- OPC Foundation. (1999). Data Access automation interface standard: *version 2.02*.
- OPC Foundation. (2001). Data Access custom interface standard: *version 2.05*.
- OPC Foundation. (2002). Alarms and Events custom interface standard: *version 1.10*.
- OPC Foundation. (2003). Data Access custom interface standard: *version 3.00*.
- OPC Foundation. (2003). OPC Historical Data Access specification: *version 1.20*.
- OPC Foundation. (2017). OPC Unified Architecture: Part 1: overview and concepts: *Release 1.04*.
- OPC Foundation. (2017). OPC Unified Architecture: Part 3: address space model: *Release 1.04*.
- OPC Foundation. (2017). OPC Unified Architecture: Part 4: services: *Release 1.04*.
- OPC Foundation. (2017). OPC Unified Architecture: Part 5: information model: *Release 1.04*.
- OPC Foundation. (2019). OPC Unified Architecture: *Interoperability for Industrie 4.0 and the Internet of Things*.
- OPC Foundation. (2020). What is OPC?, viewed 14 January 2020, <<https://opcfoundation.org/about/what-is-opc/>>.
- OPC Foundaton. (2020). Case Studies, viewed 3 April 2020, <<https://opcfoundation.org/resources/case-studies/>>.
- Postól, M. and Zbrzezny, M. (2006). Wprowadzenie do technologii OPC. CAS.
- Schwarz, M.H. and Börcsök, J. (2013). A survey on OPC and OPC-UA. In International Conference on Information, Communication and Automation Technologies (ICAT).
- Shimanuki, Y. (1999). OLE for process control (OPC) for new industrial automation systems. In 1999 IEEE International Conference on Systems, Man and Cybernetics, vol. 6, 1048-1050.
- Son, M. and Yi, M. (2010). A study on OPC Specifications: Perspective and Challenges. In International Forum on Strategic Technology, 193-197.
- Steinkrauss, U. (2010). Overview: OPC Unified Architecture: Technical overview and short description. Automation Systems Communication Laboratory.
- Thompson, D., Exton, C., Garret, L., Sajeev, A.S.M. and Watkins, D. (1997). Distributed Component Object Model (DCOM). Monash University.
- Visaya. OPC and OPC UA: windows and industrial devices, viewed 04 February 2020, <<https://visaya.solutions/en/article/opc-and-opc-ua/>>.
- Youngsheng, Z., Miaomiao, L. and Fengzhi, Z. (2013). The implementation of PLC monitoring system based on OPC. In IEEE 4th International Conference on Software Engineering and Service Science, 462-466.