Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 00 (2019) 000–000

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

47th SME North American Manufacturing Research Conference, NAMRC 47, Pennsylvania, USA

Integrating A Dynamic Simulator and Advanced Process Control using the OPC-UA Standard

Hasan Latif[a,*], Guodong Shao[b], Binil Starly[a]

[a] *Department of Industrial & Systems Engineering, The NC State University, Raleigh, NC 27606, USA*
[b] *National Institute of Standards and Technology, Gaithersburg, MD 20899, USA*

Abstract

Insufficient interoperability has long been an issue on the factory floor, however, new technologies and standards are enabling production systems to become more agile and interoperable. A communication standard can, for example, make interoperation among different vendor-specific software and hardware tools in production systems easier and more reliable. In this paper, we share our research results and experience for the establishment of a connection between a dynamic simulator and an advanced process controller in a manufacturing system using OPC-UA. The OPC-UA communication protocol, which is middleware, acts as a common interface between these systems. We established the client and server for communication and defined an exchange data structure based on the OPC-UA standard for a control problem in a chemical process plant. The case study is a proof of concept of the OPC-UA standard implementation to support interoperability for different domains.

Keywords: OPC-UA; Simulation; Advanced Process Control; Integration; Automation.

---

1. Introduction

Manufacturing industries generate huge amounts of data, however, because it is difficult to effectively exchange data among the variety of manufacturing systems and applications on the factory floor, these data typically cannot be fully exploited. A McKinsey Global Institute report states that U.S. manufacturing can maintain competitiveness by applying an optimized, autonomous factory approach in a digitized and integrated value chain [14]. Achieving such digitization and integration within a manufacturing factory is, however, heavily dependent on plant floor data and communication techniques and common semantics. A free communication flow among different software and hardware systems, in turn, typically critically relies on the proper application of standards and proven methods [12]. Exploitation of advanced technologies through application of standards is one of the major challenges facing the manufacturing industry [4].

In recent years, there has emerged a significant shift in the interconnection of physical components on the manufacturing floor where transmitted information is used for control purposes [3]. Many quintessential requirements have been identified: ubiquitous connectivity, local intelligence, safety, self-organization, flexibility, massive data monitoring, and efficiency, to name a few [10] . The most common and crucial identified factor is efficient and reliable communication when dealing with heterogeneity and interoperability of various entities on a manufacturing floor. Advanced communication and information technologies can help achieve reliable, smooth, and robust integration between manufacturing levels through various physical media and protocols [21]. This paper reports a case study that integrates a simulator and controller via communication protocol: OPC UA. OPC UA is a sophisticated, scalable and flexible mechanism for establishing secure connections between clients and servers. This paper uses Tennessee Eastman problem to formulate the base problem. [5].

This section also provides a baseline for Digital Twin in terms of system integration, automation, and control. It establishes a standard-based communication between two different platforms on a manufacturing floor. The flexible and scalable communication approach can be used for similar manufacturing problems. The rest of the section is structured as follows: Section 2 discusses the related work, Section 3 provides the details

---

[*] Corresponding author. Tel.: +1-520-330-6058 ; fax: +1-919-515-5281.
E-mail address: hhlatif@ncsu.edu (Hasan Latif).

of the case study implementation, Section 4 presents the findings, and Section 5 provides the conclusion and discussion.

2. Background

A common infrastructure model and communications framework can improve interoperability, enable more secure and efficient data transmission, and facilitate smart data usage. The research community has made significant efforts to introduce digital communications in control and field networks [7]. In this paper, we use the OPC-UA standard to enable communication and integration between the controller and the simulation of the Tennessee Eastman process. The reasons behind selecting OPC-UA over publish-subscribe technologies are multi-folds. It is platform independent, scalable, user friendly, and secure. OPC-UA is a completely new paradigm for systematic communication.

2.1. OPC-UA

Increasing demand for data exchange in a manufacturing plant requires better efficiency in communication networks [15]. As a result, newer advanced automation and control domains continue to emerge [1]. These new domains face a continuously increasing requirement of integration and interoperation. Therefore, standardized communication protocols are crucial for integrating manufacturing systems [11] [8]. OPC emerged as an automation standard primarily driven by automation vendors in process industry [19]. OPC defines a standard set of objects, interfaces, and methods to facilitate interoperability between control devices and systems. OPCs connectivity layer helps improve system interoperability.

In the 1990s, Microsoft introduced the Component Object Model (COM) and the Distributed COM (DCOM) interface standards. In 1995, Rockwell, Opto22, Intellution, and Fisher Rosemount developed a data-access standard based on COM and DCOM, and called OPC. Classical OPC include DA (Data Access), AE (Alarm & Events), HDA (Historical Data Access), and DX (Data Exchange). Each of these interfaces has a unique read and write command structure that impacts only one interface at the time. OPC-UA can be implemented on multiple platforms and no longer relies on COM/DCOM technologies [20].

The objective of the OPC-UA is to fulfill all the requirements for platform-independent system interfaces with versatile modeling capabilities that satisfy the needs of even complex systems. Independence of platform and scalability are necessary to facilitate the integration of OPC interfaces directly into a system that runs on various platforms. Access control and security are also crucial requirements because communication should be allowed through firewalls. The basic premise of OPC-UA is that the client can access small pieces of data without having to understand the entire complex model.

Therefore, it is widely accepted as an enabling technology for digital manufacturing [16]. So far, OPC-UA has been implemented by almost 20 different industrial sectors including tobacco, pharmaceuticals, and automation industries. These users have documented limitations including insufficient semantics, data models, dependence on COM/DCOM technology, inadequate security, and lack of implementation Application Programming Interfaces (API). This paper reports our effort to address these concerns by providing a case study and an implementation scenario.

2.2. Tennessee Eastman Problem

Tennessee Eastman (TE) is a well-known industry problem [5]. The original TE problem has a complex structure. It includes a three-unit operation: an exothermic, two-phase reactor; a flash separator; and a stripper. The TE problem contains 41 measured output variables and 12 manipulated variables. The TE problem has been solved with efficient algorithms using different modeling languages and tools. Downs & Vogel (1993) provided FORTRAN code of the model but did not publish the model equations. As a replacement, they provided a flow sheet, a steady-state material balance, and a qualitative description of the critical process characteristics. So, researchers who adopt the case need to make some assumptions to fulfill the missing information. In this case study, a simplified version of the TE problem has been adopted [18]. The simplified TE problem is in the steady state with a relatively modest structure. The details of the simplified TE problem will be discussed in the next Section.

3. Case Study

In this paper, we have adopted the simplified version of the TE process as the case study and performed the simulation and control modeling. We model the controller and the TE problem simulation using two different applications between which information must constantly flow. OPC-UA acts as middleware between the applications. The scenario is pragmatic and can be reused for other similar real-world cases. This case study serves as a prime example of demonstrating how OPC-UA is implemented for data communication within a plant.

Figure 1 illustrates the overview of information flow of the simplified TE process. First, we have derived an optimization problem from the simplified TE case. Then, we identify the optimal parameters for controllers. These optimal parameters, as control set points, are then sent to the process simulator via OPC-UA. Simulator sends the feedback to the controller at regular intervals. With this feedback, controller adjusts the new optimal parameters and send them back to process simulator. Therefore, a continuous and effective communication takes place at regular interval.

3.1. Simplified Tennessee Eastman Problem

As shown in Figure 2, the simplified TE problem includes a combined reactor and separator vessel. The model has two input flows (Feed 1 and Feed 2) and two output flows (Feed 3 and Feed 4). Feed 1 admits gas compounds A and C into the reactor while pure A is used to control the ratio between A and

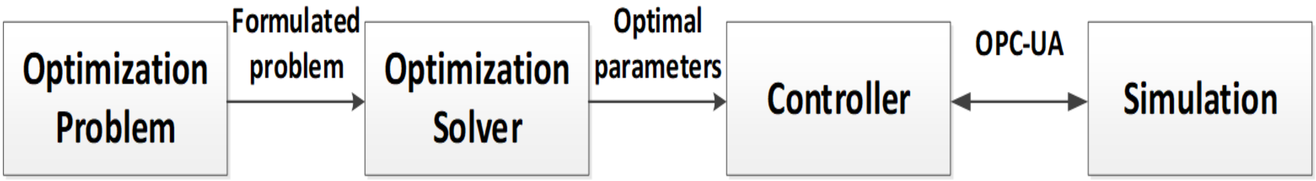


Fig. 1. Information Flow between modules of the case study

C through Feed 2. Product D, a liquid, exits through Feed 4, while the purge vapor flows out through Feed 3. In summary, the inputs of the system are A and C, and the outputs are D and the vapor purge as seen in Equation 1.

$$A + C = D + Purge \tag{1}$$

3.2. *Optimization Problem Formulation*

The optimization problem is used to derive the optimal parameter values to serve as control set points. The optimization problem has been formulated based on the model developed by Ricker (1993), which assumes that the plant is at the steady state. The optimization objective of the problem is to minimize the instantaneous cost of producing a given amount of product D per hour, which depends on three user-provided input parameters: the product flow rate in kmol per hour, the cost per kmol of A, and the cost per kmol of C. The optimization result includes optimal values for six parameters that allow users to enact the most cost-effective setup. The parameters manipulated to achieve minimum cost are the valve positions (as a percentage open) of Feeds 1 to 3 as well as the total pressure of the system. From these values, the valve position of Feed 4 can also be calculated. These five variables, as well as the instantaneous cost, are returned after the optimization execution. The mathematical model is described below:

$$\begin{aligned} Minimize\ C = 1/F_4 * [C_A(y_{A1}\chi_1 F_{1max} + \chi_2 F_{2max} - F_4) \\ + C_C(y_{C1}\chi_1 F_{1max} - F_4)] \end{aligned} \tag{2}$$

such that

$$\begin{aligned} & k_0\left(\frac{P}{\chi_3 C_{v3}\sqrt{P-100}}\right)^{1.6} * (y_{A1}\chi_1 F_{1max} + \chi_2 F_{2max} - F_4)^{1.2} \\ & * (y_{C1}\chi_1 F_{1max} - F_4)^{0.4} - F_4 \ \leq\ 0 \end{aligned} \tag{3}$$

and

$$y_{C1}\chi_1 F_{1max} \ \geq\ 0.8(y_{A1}\chi_1 F_{1max} + \chi_2 F_{2max})$$

where,

χ_1 = Feed 1 valve position (%, expressed as decimal)
χ_2 = Feed 2 valve position (%, expressed as decimal)
χ_3 = Purge valve position (%, expressed as decimal)
$\chi_4 = \frac{P}{\chi_3 C_{v3}\sqrt{P-100}}$
= Product valve position (%, expressed as decimal)
P = Total pressure of system (kPa)
F_4 = Product flow (kmol/h)
C_A = Cost of A ($/kmol)
C_C = Cost of C ($/kmol)
y_{A1}= Concentration of A in Feed 1 (%, expressed as decimal)
y_{C1} = Concentration of C in Feed 1 (%, expressed as decimal)
F_{1max} = Maximum flowrate of Feed 1 (kmol/h)
F_{2max} = Maximum flowrate of Feed 2 (kmol/h)
k_0= Constant value associated with reaction

Equation 2 represents the relationship between the reaction rate of the system and the product flow rate based on the time-based equations from the model. Since the problem was assumed to be in steady state, equation 2 was derived by setting Rickers state equations (1) through (4) equal to zero [18]. The cost equation naturally favors A, so equation 3 ensures that an ideal ratio between A and C is maintained. Table 1 lists the variables and their descriptions. The variables are sorted into three categories: output variables, input parameters, and nominal values. An optimal value is assigned the output variables by the optimization solver, input parameters, as constants, are provided by the user, nominal values are taken from Table 1 [18].

After executing the optimization, the optimal and target values (feed valves 1, 2, 3, and 4) are derived. These target values are used by the controller as set points values.

3.3. *OPC-UA Client Development: Advanced Process Control*

Process control plays a vital role ensuring conformity to process rules and protecting the process environment. Real-time optimization (RTO) can be deployed in a controller to determine the optimum control set points for the current operating conditions and constraints. The operating constraints for a plant are identified as part of the process design. During plant operations, the optimum operating conditions can change regularly owing to product throughput, process disturbances, by-product as wastes, and economic evaluations. Therefore, it is profitable to recalculate the optimum operating conditions on a regularly.

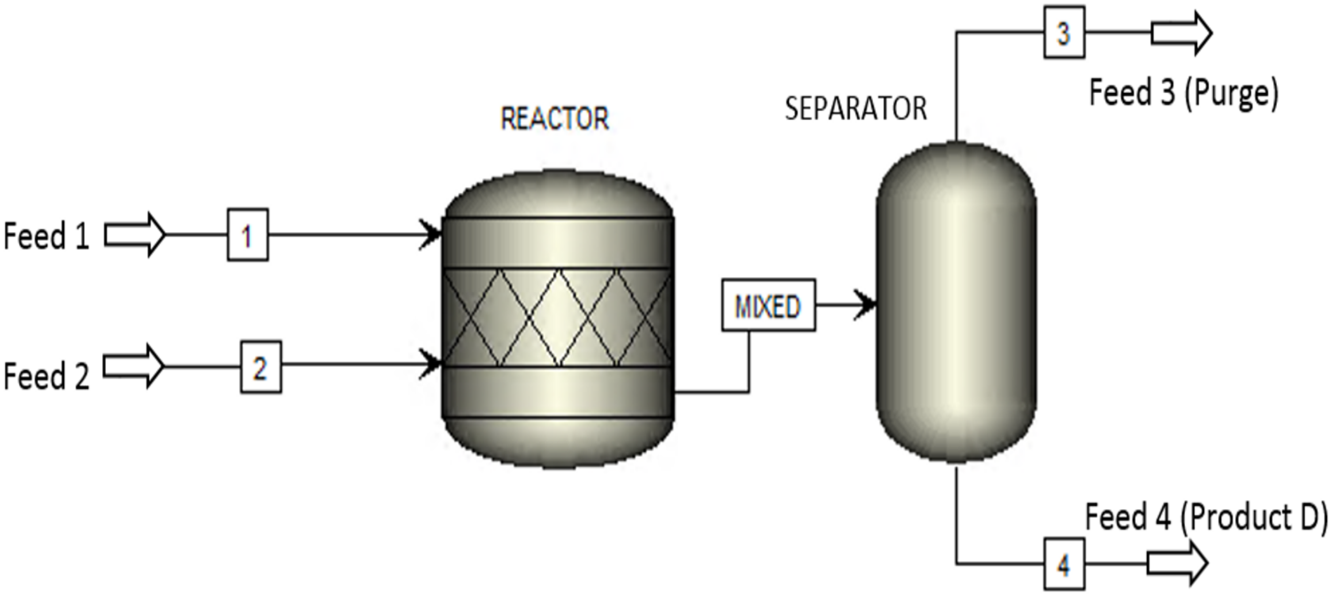


Fig. 2. Process Schematic of the Simplified TE Problem

Table 1. Summary of variables and nominal operating conditions

Output Variable	Set Value	Description	Units
χ_1	0.609533	Feed 1 valve position	(%)
χ_2	0.250223	Feed 2 valve position	(%)
χ_3	0.392578	Feed 3 valve position	(%)
χ_4	0.470302	Feed 4 valve position	(%)
P	2700	Total system pressure	kPa
C	0.2415	Instantaneous cost	\$/kmol
Manipulated Variable	**Set Value**	**Description**	**Units**
F_4	100	Product flowrate	Kmol/hour
C_A	2.206	Cost of A	\$/kmol
C_C	6.177	Cost of C	\$/kmol
Constants	**Set Value**	**Description**	**Units**
y_{A1}	0.485	Concentration of A in Feed 1	(%)
y_{C1}	0.510	Concentration of C in Feed 1	(%)
F_{1max}	330.46	Max flow rate of Feed 1	Kmol/hour
F_{2max}	22.46	Max flow rate of Feed 2	Kmol/hour
k_0	0.00117	Constant for assumed isothermic reaction	–

In this paper, a model predictive control (MPC) is designed to control the TE process simulation. A predictive model controller is part of a multi-level control hierarchy in modern processing plants [21]. We use Aspen DMC3 to develop the MPC controller [9].

Three different types of variables are used: manipulated (MV), controlled (CV), and disturbance variables (DV). The three manipulated variables are three valve positions: U_1, U_2, and U_3 respectively. The three controlled variables are product flowrate F_4, pressure (P), and product A in the by-product Y_{A3}.

The relationship between controlled variables and manipulated variables are adapted from Ricker (1993) [18]. The connections are derived in transfer function format from a state space model of the TE problem.

$$y = \begin{bmatrix} F_4 \\ P \\ y_{A3} \end{bmatrix} = Gu = \begin{bmatrix} g_{11} & 0 & 0 \\ g_{21} & 0 & g_{23} \\ 0 & g_{32} & 0 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} \tag{4}$$

$$g_{11} = \frac{1.7}{0.75s + 1} \tag{5}$$

$$g_{21} = \frac{45(5.67s + 1)}{2.5s^2 + 10.25s + 1} \tag{6}$$

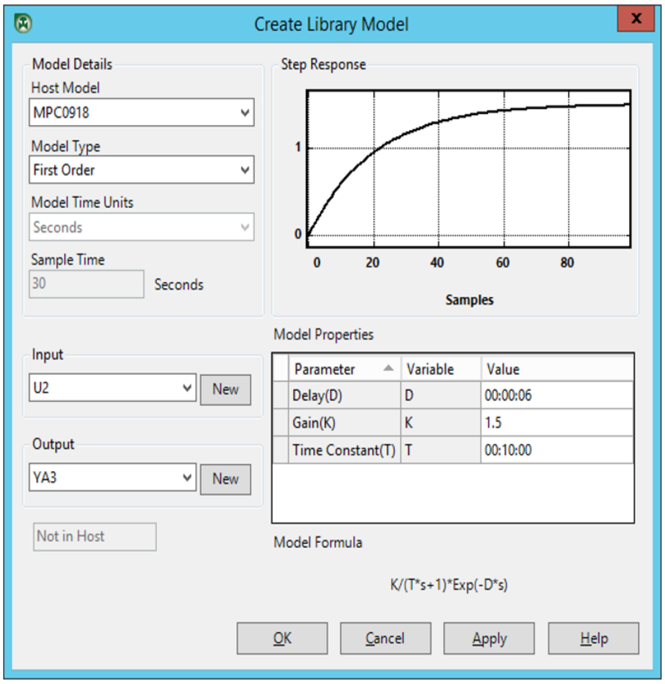
$$g_{23} = \frac{-23.81s - 2.086}{s^2 + 7.874s + 0.1915} \tag{7}$$

$$g_{32} = \frac{1.5}{10s + 1} e^{-0.1s} \tag{8}$$

The constraints of the model are given as below.

- Pressure (P) has an upper bound (3000 Kpa).
- A in the purge $yA3$ has a range (0.429 ¡ $yA3$ ¡ 0.886).
- Product flow F_4 has a set point (100 Kmol/hr).
- All threee manipulated variables are unconstrained.

Using the transfer functions, library models are created in Aspen DMC3. Different types of state space models can be stored in a library. These library models can be reused to establish the relation between manipulated and controlled variables. For instance, the first order transfer functions library model formula is $\frac{K}{T*s+1}e^{-D*s}$
where T = Time Constant, D = Delay, K = Gain.
In this problem, transferring g32 to model library provides T = 10 mins; K = 1.5; D = 6 sec.
Aspen DMC3 provides a visual representation of the library model as well. The graph of transfer function g_{32} is given in Figure 3.



Model Details
Host Model: MPC0918
Model Type: First Order
Model Time Units: Seconds
Sample Time: 30 Seconds
Input: U2
Output: YA3
Not in Host

Model Properties

Parameter	Variable	Value
Delay(D)	D	00:00:06
Gain(K)	K	1.5
Time Constant(T)	T	00:10:00

Model Formula
K/(T*s+1)*Exp(-D*s)

Fig. 3. Transfer Function Graph of a Model Predictive Controller (MPC)

After storing all the transfer functions in the library model, a master model is prepared. After being simulated offline, the controller is ready to deploy.

3.4. OPC-UA Server Development: Simulation

A Modelica model has been developed [13] to simulate the dynamic behavior of the simplified TE process, and it is based on the mathematical description provided by Ricker (1993). This Modelica model library includes the model of the open loop plant. It consists of a model named Reactor, a connector designated pCon, models for setting the boundary conditions, and two models describing the input and output source valves. The Reactor model represents the processing unit that combines the behavior of the reactor and the separator. These models have been used to compose the ReactorOpenLoop model, a model describing the behavior of the open loop plant. The model library is written in Modelica 3.3 and has been tested using Dymola 2018 and OpenModelica 1.11.0 64 bits under Windows 2010. The model has been used for the ISO 15746 standard implementation.

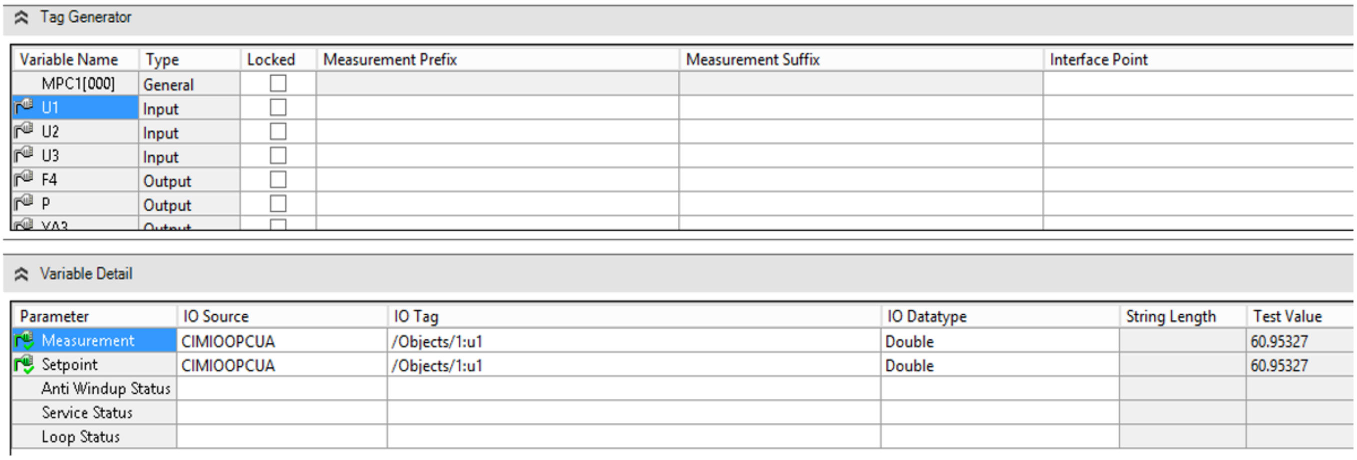
3.5. Communication Protocol: OPC-UA

The Modelica simulation, discussed in previous subsection, is acting as an OPC server. The controller, designed in Aspen DMC3, is acting as an OPC client.
To make a successful OPC-UA connection, OPC client and server need to communicate via nodes. The OPC server needs to identify the nodes and read data successfully. To setup an OPC client using Aspen DMC3, the Cim-IO interface manager first needs to be started. The CIM-IO interface is a communication interface that provides a communication standard for interfacing with various AspenTech products like InfoPlus.21 and third-party software such as Modbus, OPC servers. Through DMC3s CIM-IO interface manager, the OPC-UA interface gets active and ready to communicate with the server.
Next, the OPC-UA client requires connecting to the OPC-UA server via nodes. The nodes addresses are provided in the modeling. Then the OPC-UA connection via nodes is tested and deployed. Figure 4 is a screen capture that shows the variable names and types, as well as the node address assignment.
After a successful OPC-UA server/client connection, the controller MPC1 is deployed and starts running. In the Aspen Web Interface module, the feedback from controller and simulation is observable. The history, data exchange information, and controller application can be seen and changed from this module according to the user need.

The optimization execution result provides the controller with target set values for feed valve positions. The controller uses the constraints to acquire real-time feed valve positions and communicates with the simulation through the OPC-UA messaging protocol. The simulation also provides feedback to the controller via OPC-UA and the controller acts as a check and balance element in the simulation by providing the next set of real-time feed valve positions. Overall, this case study is an implementation of a standard-based communication protocol in the manufacturing domain. The approach can be applied to similar problems in the plant to enable real-time communication between different enterprise levels. Automotive, medical device, consumer electronics, aerospace & defense industry can adapt the technology and march towards digital manufacturing.

4. Lessons Learned

Even though in this case study, the OPC-UA has been successfully implemented between a controller and a simulator of



Variable Name	Type	Locked	Measurement Prefix	Measurement Suffix	Interface Point
MPC1[000]	General	☐			
U1	Input	☐			
U2	Input	☐			
U3	Input	☐			
F4	Output	☐			
P	Output	☐			
YA3	Output	☐			

Parameter	IO Source	IO Tag	IO Datatype	String Length	Test Value
Measurement	CIMIOOPCUA	/Objects/1:u1	Double		60.95327
Setpoint	CIMIOOPCUA	/Objects/1:u1	Double		60.95327
Anti Windup Status					
Service Status					
Loop Status					

Fig. 4. Test and Deployment of Node Address

the simplified TE problem, OPC-UA implementation requires some complicated procedures. Multiple challenges need to be addressed in a proficient manner to have a smooth OPC-UA implementation. We have identified the following such major challenges based on our experience.

- The challenge of selecting OPC-UA server/client enabled applications and specification of a well-defined architectures: not all the control and simulation applications are OPC-UA enabled, so effort is needed to select an application that is not only capable of modeling the problem (e.g., control) but also establishing a server or a client. Also, OPC-UA has a large set of specifications. It is difficult to assess and estimate the project effort and development time in the beginning of the project. The existing physical system has a limited capacity that can sometimes hinder additional functionality. For instance, the existing system has only a limited amount of RAM space or processor clockwork speed available for additional OPC-UA accommodation. As OPC-UA memory utilization increases, it poses a threat to the existing infrastructure to crash.
- Because OPC-UA connects a multitude of applications across firewalls and networks, server security becomes a concern. Like many other message protocol systems, OPC-UA uses authentication, authorization, and encryption via an address space concept. OPC-UA address space provides a standard way for servers to represent objects to clients. It defines objects in terms of variables and methods. The elements of a model are represented in the address space as nodes that are assigned to a node class, e.g., objects, variables, and methods. On the other hand, the software components have different levels of maturity for creating the address space model. Some of these software components provide a graphical user interface (GUI) to model address space and to add nodes and references, while others do not. The GUI generates code to establish OPC-UA connection between servers and clients. With a GUI, therefore, it will be easier for building up the server/client connection. Although the node address can be identified from literature review, experiments, or manuals, none of these methods is very intuitive. For instance, in this case study, we have identified the OPC-UA server node address via a reference and the OPC-UA client node address by setting up an additional test server. With the help of the test server, the connection is established with the OPC-UA client and the node address is captured through the connection details.
- An OPC-UA server contains sets of services that are used by the clients. All the OPC-UA functionalities exist in these services, which have a request and response message. Services are defined in the OPC-UA standard and the user cannot change them. For instance, while implementing the OPC-UA communication between server and client, it is very important to ensure that each follows the same (32 bit/64bit) communication bits. Users cannot modify any of the system-defined architecture.
- Security protocol is pre-defined in OPC-UA. OPC-UA claims all the required security features are built in to minimize the efforts from the developers [2]. In this case study, certificate authority is used as a security measure to ensure data protection. A Certificate Authority creates and verifies certificates. It also adds a digital signature to the Certificate to confirm user identity. OPC UA applies different security tiers. It is found that security tiers largely depends on the application platform. User has little to none control over choosing the security tier once the application platform is chosen.
- There is no way to measure the performance of the OPC-UA connection currently. At any point of OPC-UA establishment, it is hard to understand the level of reliability and quality of connection. Moreover, OPC-UA cannot determine data quality which impacts the performance as well. Should it become possible to quantify or approximate the performance of the OPC-UA connection, different methods could apply, e.g., memory rearrangement, structural reallocation, existing firmware upgrades.

- Finally, to accomplish a complete semantics interoperability, OPC-UA alone is not sufficient because it only enables syntactic interoperability between clients and servers. There is a great need for semantics to support analytics and scalability across various application from different vendors.

Overall, these key challenges contribute to the cost of OPC-UA implementation and create additional uncertainties.

5. Discussion and Conclusion

Interoperability is a very critical issue that manfuacturers have to deal with. Communication standards such as OPC-UA make it possible to have efficient and reliable information exchange between enterprise levels. By integrating with other manufacturing interoperabilty standards with semantics such as MTConnect e.g., MTConnect-OPC-UA cmpanion specification, OPC-UA could become an important piece of in semantic interoperability for industrial applications. Even though OPC UA provides the largest eco-system for industrial operability, OPC foundation has unveiled a new version of OPC-UA called OPC-UA PubSub [6][17]. PubSub enables the use of OPC UA directly over the Internet by utilizing popular data transports like MQTT and AMQP. At the same time, it retains key OPC UA end-to-end security and standardized data modeling advantages.

This paper presents an approach for implementing OPC-UA as middleware between different manufacturing systems. The integration of process simulation and advanced process control from two different application environments has not been done before. The case study identified the implementation requirements for the applied problem, standards, and technologies. The feasibility of scenario is also verified in the case study. Valuable lessons learned have been discussed.

However, in this work, the OPC-UA technology was tested with only a small stream of data in a laboratory environment. In a real-work application, enormous amounts of data have to be transferred and exchanged, which may complicate the implementation. The performance of the OPC-UA data exchange needs to be studied more closely. There are several ways that this work could be continued in the future. A manufacturing case study in which more vendors products are involved for integration; a real process that generates more complex data and more realistic amounts of data could be used to replace the process simulator. In addition, the methodology and scenario of the implementation could be enhanced. More functionalities could be implemented within the system.

Disclaimer

No approval or endorsement of any commercial product by NIST is intended or implied. Certain commercial software systems are identified in this paper to facilitate understanding. Such identification does not imply that these software systems are necessarily the best available for the purpose.

References

[1] Alcaraz, C., Roman, R., Najera, P., & Lopez, J. (2013). Security of industrial sensor network-based remote substations in the context of the internet of things. Ad Hoc Networks, 11(3), 1091-1104.

[2] Armstrong, R. & Hunkar, P. (2010). The OPC UA security model for administrators. White paper V1.00, OPC Foundation. Link: `https://opcfoundation.org/wp-content/uploads/2014/05/OPC-UA_Security_Model_for_Administrators_V1.00.pdf`

[3] Bradley, D., Russell, D., Ferguson, I., Isaacs, J., MacLeod, A., & White, R. (2015). The Internet of ThingsThe future or the end of mechatronics. Mechatronics, 27, 57-74.

[4] Cwikla, G., & Foit, K. (2017). Problems of integration of a manufacturing system with the business area of a company on the example of the Integrated Manufacturing Systems Laboratory. In MATEC Web of Conferences (Vol. 94, p. 06004). EDP Sciences.

[5] Downs, J. J., & Vogel, E. F. (1993). A plant-wide industrial process control problem. Computers & chemical engineering, 17(3), 245-255.

[6] Drahos, P., Kucera, E., Haffner, O., & Klimo, I. (2018, January). Trends in industrial communication and OPC UA. In 2018 Cybernetics & Informatics (K&I) (pp. 1-5). IEEE.

[7] Espi-Beltran, J. V., Gilart-Iglesias, V., & Ruiz-Fernandez, D. (2017). Enabling distributed manufacturing resources through SOA: The REST approach. Robotics and Computer-Integrated Manufacturing, 46, 156-165.

[8] Fruhwirth, T., Pauker, F., Fernbach, A., Ayatollah, I., Kastner, W., & Kittl, B. (2015, November). Guarded state machines in OPC UA. In Industrial Electronics Society, IECON 2015-41st Annual Conference of the IEEE (pp. 004187-004192). IEEE.

[9] Golightly, R. 2018. The Aspen DMC3 Difference. White paper.

[10] Gonzalez, I., Calderon, A. J., Barragan, A. J., & Andujar, J. M. (2017). Integration of Sensors, Controllers and Instruments Using a Novel OPC Architecture. Sensors, 17(7), 1512.

[11] Hernandez, L., Baladron, C., Aguiar, J. M., Calavia, L., Carro, B., Sanchez-Esguevillas, A., & Gomez, J. (2012). A study of the relationship between weather variables and electric power demand inside a smart grid/smart world framework. Sensors, 12(9), 11571-11591.

[12] Lu, Y., Morris, K. C., Frechette, S. (2016). Current Standards Landscape for Smart Manufacturing Systems. NISTIR 8107.

[13] Martin-Villalba, C., Urquia, A., & Shao, G. (2018). Implementations of the Tennessee Eastman Process in Modelica. In Proceeding of the 9th Vienna International Conference on Mathematical Modeling. Elsevier, IFAC-PapersOnLine. Volume 51. Issue 2. Pages 619-624.

[14] Ramaswamy, Sree, J. Manyinka, G. Pinkus, K. George, J. Law, T. Gambell, and A. Serafino. ”Making it in America: Revitalizing US manufacturing.” New York, NY: McKinsey Global Institute. Google Scholar (2017).

[15] Mejias, A., Herrera, R. S., Marquez, M. A., Calderon, A. J., Gonzalez, I., & Andujar, J. M. (2017). Easy Handling of Sensors and Actuators over TCP/IP Networks by Open Source Hardware/Software. Sensors, 17(1), 94.

[16] Pauker, F., Frhwirth, T., Kittl, B., & Kastner, W. (2016). A systematic approach to OPC UA information model design. Procedia CIRP, 57, 321-326.

[17] OPC Foundation Press Release, March, 2018. Link: `https://opcfoundation.org/news/press-releases/opc-foundation-announces-opc-ua-pubsub-release-important-extension-opc-ua-communication-platform/`

[18] Ricker, N. L. (1993). Model predictive control of a continuous, nonlinear, two-phase reactor. Journal of Process Control, 3(2), 109-123.

[19] Rohjans, S., Piech, K., & Lehnhoff, S. (2013, November). UML-based modeling of OPC UA address spaces for power systems. In Intelligent Energy Systems (IWIES), 2013 IEEE International Workshop on (pp. 209-214). IEEE.

[20] Toivanen, S. (2016). Implementation of performance assessment tool for multivariable, model-predictive controller. MS Thesis, Tampere University of Technology.

[21] Wagner, T. (2003, September). Applying agents for engineering of industrial automation systems. In German Conference on Multiagent System Technologies (pp. 62-73). Springer, Berlin, Heidelberg.