

**REALIZATION OF PLC CONTROLLER-TO-CONTROLLER
VERTICAL COMMUNICATION BY USING OPC-UA**

LIEW JIA XIAN

**BACHELOR OF MECHATRONICS ENGINEERING WITH
HONOURS
UNIVERSITI TEKNIKAL MALAYSIA MELAKA**

2019

**REALIZATION OF PLC CONTROLLER-TO-CONTROLLER VERTICAL
COMMUNICATION BY USING OPC-UA**

LIEW JIA XIAN

**A report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Mechatronics Engineering with Honours**

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

DECLARATION

I declare that this thesis entitled “REALIZATION OF PLC CONTROLLER-TO-CONTROLLER VERTICAL COMMUNICATION BY USING OPC-UA is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : _____
Name : _____
Date : _____

APPROVAL

I hereby declare that I have checked this report entitled “REALIZATION OF PLC CONTROLLER-TO-CONTROLLER VERTICAL COMMUNICATION BY USING OPC-UA” and in my opinion, this thesis it complies the partial fulfillment for awarding the award of the degree of Bachelor of Mechatronics Engineering with Honours

Signature :

Supervisor Name :

Date :

DEDICATIONS

To my beloved mother and father

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude and appreciation to my project supervisor, Dr. Saifuzal Bin Alwi for his guidance, encouragement and advises over this project. This project would not able to process and accomplished on the right track without his assistance and helpful advice.

I would also like to take this opportunity to thank Universiti Teknikal Malaysia Melaka (UTeM) for providing this valuable opportunity and facilities to carry out this meaningful final year project. I am also very thankful to the lab assistant Mr. Norhisham bin Abu Seman for allowing me to use the laboratories inside the lab and his kindness helps during my works inside the lab.

Last but not least, I would like to express my thanks to all my family and fellow friends who give support and helps along with this project.

ABSTRACT

Programmable Logic Controller (PLC) plays an important role in the automation process in the manufacturing industries. Typically, more than one type of PLC is used in a complete manufacturing automation process. Supervisory and control of every process in a manufacturing process are needed in modern industries to ensure a smooth and efficient manufacturing production process. Interoperability standard is needed to achieve the communication between the PLCs in the field layers with the upper supervisory level. OPC-UA is a platform independently interoperability standard for the open source controller and devices. The objectives of this report are to determine the function of client-server in OPC-UA implementation, to establish the architecture of OPC-UA for vertical communication between PLC and Upper-level controller devices or the SCADA system using KEPServerEX OPC-UA server, and to analyze the performance of OPC-UA in the communication. A simple OPC-UA server and client had been developed to visualize the working principle of OPC-UA client-server by using Node.js. The server is also connected to the UaExpert OPC-UA client to testing and verify the developed server. Vertical communication between the Omron CP1E PLC and Ignition SCADA system through KEPServerEX OPC-UA server with the Omron Fins Ethernet connection had been established and presented. The data of the PLC in the communication also being observed using UaExpert OPC-UA client. Experiments had been set-up to analyze the performance of OPC-UA architecture in communication by measuring the average time taken to complete the READ/WRITE instruction to the data tags under different hardware specification and security profile configuration by using the performance tools of UaExpert OPC-UA client.

ABSTRAK

Pengawal Logik yang Boleh Diprogram (PLC) memainkan peranan penting dalam proses automasi dalam industri perkilangan. Biasanya, lebih daripada satu jenis PLC digunakan dalam satu proses automasi pembuatan lengkap. Pengawasan dan pengendalian setiap proses dalam proses pembuatan diperlukan dalam industri yang moden untuk memastikan proses pengeluaran pembuatan yang licin dan efisien. Standard interoperabilitas diperlukan untuk mencapai komunikasi antara PLC dalam lapisan lapangan dengan lapisan penyeliaan atas. OPC-UA adalah satu standard platform interoperabilitas bebas untuk pengawal dan peranti sumber terbuka. Objektif laporan ini adalah untuk menentukan fungsi klien-pelayan dalam pelaksanaan OPC-UA, menubuhkan arsitektur OPC-UA untuk komunikasi menegak antara PLC dan peranti pengawal peringkat atas sistem SCADA dengan menggunakan perisian OPC-UA KEPServerEX dan menganalisis prestasi OPC-UA dalam komunikasi. Sepasang pelayan dan klien OPC-UA yang ringkas telah dibangunkan untuk memvisualisasikan prinsip kerja OPC-UA client-server dengan menggunakan node.js. Pelayan tersebut juga disambungkan dengan klien OPC-UA UaExpert untuk mengujikan dan mensahkan pelayan tersebut. Komunikasi antara sistem Omron CP1E PLC dan Ignition SCADA melalui KEPServerEX OPC-UA pelayan dan Omron Fins Ethernet telah distabilkan dan dibentangkan. Eksperimen-eksperimen telah dijalani untuk menganalisis prestasi OPC-UA dalam komunikasi dan hasilnya telah dibincangkan. Prestasi OPC-UA dalam komunikasi dinilai dengan mengukur masa yang diperlukan untuk habiskan arahan READ/WRITE pada data tags yang lokasi pada mesin yang mempunyai spesifikasi dan sekuriti profil yang berbeza dengan menggunakan alat menguji prestasi pada UaExpert OPC-UA klient.

TABLE OF CONTENTS

	PAGE
DECLARATION	
APPROVAL	
DEDICATIONS	
ACKNOWLEDGEMENTS	2
ABSTRACT	3
ABSTRAK	4
TABLE OF CONTENTS	5
LIST OF TABLES	7
LIST OF FIGURES	8
LIST OF SYMBOLS AND ABBREVIATIONS	12
LIST OF APPENDICES	13
CHAPTER 1 INTRODUCTION	14
1.1 Motivation	14
1.2 Problem Statements	15
1.3 Objectives	16
1.4 Scopes	17
CHAPTER 2 LITERATURE REVIEW	18
2.1 Theory	18
2.1.1 Data Communication	18
2.1.2 Network Topologies	18
2.1.3 Interprocess Connection	19
2.1.4 OPC (Open Platform Communication)	19
2.1.5 OPC-UA (OPC Unified Architecture)	20
2.1.6 PROFINET	21
2.1.7 Comparison between OPC-UA and PROFINET	21
2.2 Research and Study	22
2.2.1 Performance of OPC/OPC-UA	22
2.2.2 Communications between OPC and PLC	24
2.2.3 Data Communication between Programmable Logic Controller	27
2.2.4 Comparison between OPC-UA and OPC Classic	29
2.3 Summary	30
CHAPTER 3 METHODOLOGY	31
3.1 Introduction	31
3.2 Project Flow	32

3.3	Software	33
3.4	OPC-UA Server and Client Development	35
	3.4.1 Node.js based OPC-UA Server	35
	3.4.2 Node.js based OPC-UA Client	37
3.5	Vertical Communication between the PLC and SCADA (HMI) by using KEPServerEX	38
	3.5.1 The Architecture of the Connection	38
	3.5.2 Connect and Upload PLC Program to the Omron CP1E-NA PLC through CX-Programmer	39
	3.5.3 Connect PLC and KEPServerEX OPC-UA Server	41
	3.5.4 Connect Ignition SCADA System and KEPServerEX OPC-UA Server	46
	3.5.5 Create a SCADA Project by using Ignition Designer	50
	3.5.6 Connect KEPServerEX OPC-UA Server to UaExpert OPC-UA Client	52
	3.5.7 Stamping System Monitoring and Controlling using OPC-UA	54
3.6	Performance Analysis of OPC-UA Communication	54
	3.6.1 Performance Analysis of the Stamping System	54
	3.6.2 Performance Analysis of OPC-UA using UaExpert	55
3.7	Complete Flowchart for OPC-UA Communication	56
CHAPTER 4 RESULTS AND DISCUSSIONS		58
4.1	Introduction	58
4.2	Results	58
	4.2.1 OPC-UA Server and Client	58
	4.2.2 Vertical Communication of Ignition SCADA and Omron CP1E PLC using KEPServerEX OPC-UA Server	60
	4.2.3 Omron CP1E PLC Data Monitoring by using UaExpert	62
	4.2.4 Stamping System Monitoring and Controlling using OPC-UA	63
	4.2.5 Performance Analysis of the Stamping system	65
	4.2.6 Performance Analysis of OPC-UA Communication using UaExpert	67
4.3	Discussion	76
CHAPTER 5 CONCLUSION AND RECOMMENDATIONS		78
5.1	Conclusion	78
5.2	Future Work	78
REFERENCES		80
APPENDICES		83

LIST OF TABLES

Table 2.1	Summary of paper related to OPC-UA performance analysis	24
Table 2.2	Summary of papers related to OPC-UA communication	27
Table 2.3	Comparison between OPC-UA and OPC Classic	29
Table 3.1	Client's Hardware Specification for Experiment 1	55
Table 3.2	Summary of Hardware Specification for Experiment 2	56
Table 4.1	Results of delay measured using the stopwatch	66
Table 4.2	Actual time taken of RED indicator light to turn off in Ignition	66
Table 4.3	Average time taken for 2 nodes to complete instruction for 100 cycles in the first Experiment (ms)	68
Table 4.4	Average time taken for 5 nodes to complete instruction for 100 cycles in the first Experiment (ms)	69
Table 4.5	Average time taken for 10 nodes to complete instruction for 100 cycles in the first Experiment (ms)	70
Table 4.6	Average time taken for 2 nodes to complete instruction for 100 cycles in the second Experiment (ms)	72
Table 4.7	Average time taken for 5 nodes to complete instruction for 100 cycles in the second Experiment (ms)	73
Table 4.8	Average time taken for 10 nodes to complete instruction for 100 cycles in the second Experiment (ms)	74

LIST OF FIGURES

Figure 1.1 Global PLCs by Region – 2017 Revenues and CAGR [2]	15
Figure 2.1 Network Topologies [8]	18
Figure 2.2 Overview of OPC Family [10]	19
Figure 2.3 OPC-UA Server/Clients Architecture [12]	20
Figure 2.4 OPC-UA Specification [13]	20
Figure 2.5 Advantages of Working with PROFINET at IO layer [15]	21
Figure 2.6 Network Hierarchy of OPC-UA and PROFINET [16]	21
Figure 2.7 Comparison between PROFINET and OPC-UA [16]	22
Figure 2.8 Real-time communication flow diagram between MATLAB and PLC based on OPC [24]	25
Figure 2.9 System Overview of paper [26]	26
Figure 2.10 Architecture of the implemented system [26]	26
Figure 2.11 Structure of the system [29]	28
Figure 2.12 Flow chart of the SC [29]	28
Figure 2.13 System Hardware Chart of [30]	29
Figure 3.1 Flowchart of the Project Flow	32
Figure 3.2 Logo of Node.js	33
Figure 3.3 UaExpert Launching Window	33
Figure 3.4 Logo of KEPServerEX	34
Figure 3.5 CX-Programmer Launching Window	34
Figure 3.6 Logo of Ignition Platform	35
Figure 3.7 Command to create a folder for OPC-UA server and add “node- opcua” library	36

Figure 3.8 Script to create the OPC-UA server instant	36
Figure 3.9 Command to initialize the server	36
Figure 3.10 Commands to create a project and include library	37
Figure 3.11 Scripts for the library, variable, and endpoint URL declaration	37
Figure 3.12 Communication architecture	38
Figure 3.13 CX-Programmer PLC configuration wizard	39
Figure 3.14 Network settings of PLC in CX-Programmer	40
Figure 3.15 Connecting the PLC and CX-Programmer	40
Figure 3.16 Transferring the PLC program to PLC	40
Figure 3.17 Connection between Omron CP1E and Server machine using Ethernet Cable	42
Figure 3.18 Identifying the IP address of the Omron CP1E PLC	42
Figure 3.19 Configuration of Ipv4 addressing	43
Figure 3.20 User Manager of KEPServerEX	43
Figure 3.21 OPC-UA server endpoint	44
Figure 3.22 Add Channel Wizard of KEPServerEX	44
Figure 3.23 PLC model selection in KEPServerEX	45
Figure 3.24 Specifying the IP address of Omron CP1E	45
Figure 3.25 Defined CP1E data tag in KEPServerEX	46
Figure 3.26 Ignition Gateway Control Utility	47
Figure 3.27 Main page of Ignition SCADA gateway	47
Figure 3.28 OPC-UA server configuration in Ignition Gateway	48
Figure 3.29 Discover OPC-UA server through OPC-UA endpoints	48
Figure 3.30 New OPC-UA connection settings in Ignition	49
Figure 3.31 OPC-UA Configuration Manager of KEPServerEX	49

Figure 3.32 OPC Server Connection in Ignition	50
Figure 3.33 New Project Setup Wizard of Ignition Designer	51
Figure 3.34 Window type selection panel in Ignition Designer	51
Figure 3.35 Add KEPServerEX data tag into Ignition Designer	52
Figure 3.36 Create user interfaces of the data tag	52
Figure 3.37 OPC-UA server discovery	53
Figure 3.38 Security settings of the OPC-UA server	53
Figure 3.39 Electro-pneumatic Stamping System	54
Figure 3.40 Complete flowchart for OPC-UA vertical communication	57
Figure 4.1 The output screen of the command window after the server is initialized	58
Figure 4.2 Output screen of command window when the client is started	59
Figure 4.3 Variables of server monitored by using UaExpert	59
Figure 4.4 An Omron CP1E PLC with Ethernet Module	60
Figure 4.5 The state of CP1E and Ignition SCADA before switch 1 is triggered	61
Figure 4.6 The state of CP1E and Ignition SCADA after switch 1 is triggered	61
Figure 4.7 Screenshot of Ignition SCADA running on a mobile device	62
Figure 4.8 Screenshot of Ignition SCADA running on a Windows 10 laptop	62
Figure 4.9 UaExpert that running on a different computer connected to the same OPC-UA server	63
Figure 4.10 Initial State of the Stamping System	63
Figure 4.11 Initial state of the Stamping system	64
Figure 4.12 State of the green indicator after START button pressed	64
Figure 4.13 State of the stamping system after START button pressed	65

Figure 4.14 Actual time taken for the RED light to turn off in Ignition	67
Figure 4.15 Average time taken for 2 nodes to complete instruction for 100 cycles in the first Experiment (ms)	68
Figure 4.16 Average time taken for 5 nodes to complete instruction for 100 cycles in the first Experiment (ms)	69
Figure 4.17 Average time taken for 10 nodes to complete instruction for 100 cycles in the first Experiment (ms)	70
Figure 4.18 Average time taken for 2 nodes to complete instruction for 100 cycles in the second Experiment (ms)	75
Figure 4.19 Average time taken for 5 nodes to complete instruction for 100 cycles in the second Experiment (ms)	75
Figure 4.20 Average time taken for 10 nodes to complete instruction for 100 cycles in the second Experiment (ms)	76

LIST OF SYMBOLS AND ABBREVIATIONS

PLC	-	Programmable Logic Controller
HMI	-	Human Machine Interface
I/O	-	Input/Output
SCADA	-	Supervisory Control and Data Acquisition
RTU	-	Remote Terminal Unit
OPC	-	Open Platforms Communication
OPC-UA	-	OPC Unified Architecture
IP	-	Internet Protocol
GACR	-	Compound Annual Growth Rate

LIST OF APPENDICES

APPENDIX A I/O MEMORY AREAS OF OMRON CP1E-NA	83
APPENDIX B PLC LADDER PROGRAM	84
APPENDIX C LADDER PROGRAM FOR STAMPING SYSTEM	85
APPENDIX D SCRIPT OF DEMO SERVER	87
APPENDIX E SCRIPT OF DEMO CLIENT	89
APPENDIX F PROJECT GANTT CHART	92

CHAPTER 1

INTRODUCTION

1.1 Motivation

A Programmable Logic Controller (PLC) is always related when we are talking about industrial automation. A Programmable Logic Controller (PLC) is a reprogrammable special form of a microcontroller-based controller that can be used by users with limited knowledge of computers to control the machines and process [1].

The usage of PLC in the process industries such as oil and gas, food and beverage, mining and etc. to automate the process are getting higher and higher over the years. According to the PLCs Report 2017 by the IHS Markit, the global PLCs market including the smart relays, various type of PLCs, software and the services, is reaching \$8.5 billion in the year 2017 and is expected to exceed \$ 9.5 billion by the year 2021 [2]. From the report, we can see that the market of PLCs is still having the potential to grow in the coming future.

Smart Factories or industry 4.0 production system is one of the core ideas of the concept of Industry 4.0 (I4.0). According to the Global Programmable Logic Controller (PLC) Market 2018-2022, the increasing requirement of PLCs in smart factories is one of the factors that lead to the growth of the PLC market in the period of 2018 to 2022 [3].

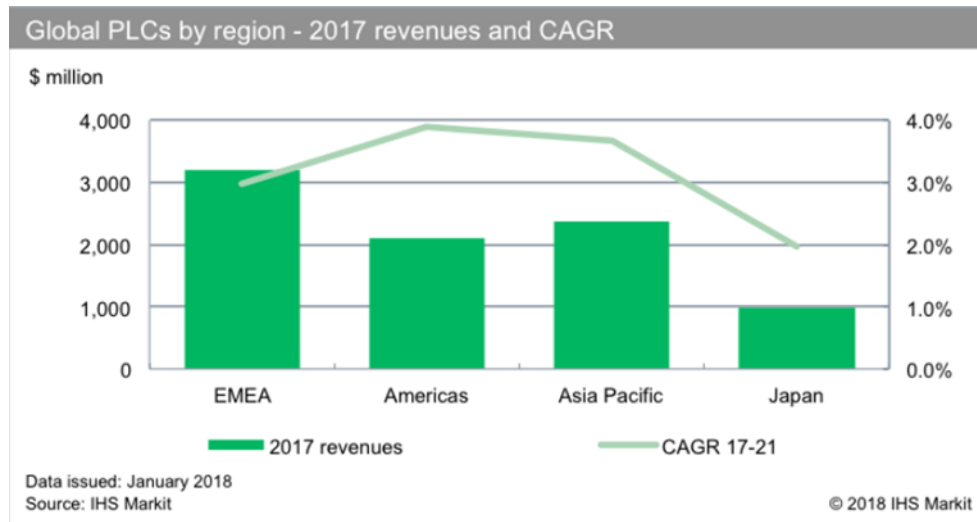


Figure 1.1 Global PLCs by Region – 2017 Revenues and CAGR [2]

Interoperability and connectivity are important elements in industry 4.0 concept [4]. Since there are many PLC vendors in the global PLC market such as Siemens, Omron, Mitsubishi Electric, Schneider Electric, etc. Therefore, in order to make sure PLCs are able to achieve the requirement of the market demands, PLCs from different vendors must be able to achieve the interoperability between machines or devices.

OPC-UA is an industrial interoperability standard to define the specifications of the PLCs communications. Therefore, it motivates me to determine the function of OPC-UA in PLCs communications, establish a vertical communication between PLCs and analyzed the performance of OPC-UA in data communications.

1.2 Problem Statements

PLCs are widely used in industrial automation since many years ago. With the aids of the PLCs, the automation processes or machines can be control and monitor by the operator. An industrial automation process is often involved of more than one PLC in the process. In most situation, these PLCs might not be the same and sometimes it might be from different PLC vendors or manufacturers. Hence an interoperability standard is needed to co-operative and communicate between these PLCs.

OPC is an industrial interoperability standard that developed and maintained by the OPC Foundation. OPC consists of a series of specifications that developed by the industry vendors and the end-users as well, which defined the interface of the Clients-Servers and the Servers-Servers including data acquisition and monitoring [5].

OPC-UA is the next generation of the conventional OPC, OPC-UA is platform independent architecture that integrated all function of the conventional OPC and provides more secure, extensible and able to model the complex information compared to the conventional OPC [6].

Human Machine Interface (HMI) is part of the SCADA (Supervisory control and data acquisition). HMI is a graphical user interface that connects persons with the device, machine, and system. HMI can be used to visualize, tracking, monitor and more in industry setting [7]. Therefore, HMI can be considered as parts of the supervisory level of a process control system.

There are some levels in a complete process control system such as field level, supervisory level, and control level. In order to realize the vertical communication between the PLC controllers (field level) and the supervisory level, OPC-UA is used to acts as medium or standard to establish communication between both levels. Through this research, a simulation will be conducted to realize the communication between the SCADA and the PLC by using the OPC-UA standard. Performance of using OPC-UA will be tested throughout this research.

1.3 Objectives

The objectives are:

1. To determine the function of client-server in OPC-UA implementation.
2. To establish the architecture of OPC-UA for vertical communication between PLC and SCADA system by using Kepware OPC-Server.
3. To analyze the performance of OPC-UA in PLC Communication.

1.4 Scopes

The scopes are:

1. Compare the OPC-UA with OPC Classic and PROFINET.
2. Develop an OPC-UA Server and Client by using Node.js
3. Verify the developed OPC-UA Server by connecting it to the OPC-UA Client and UAExpert OPC-UA client.
4. Set-up a communication between PLC and the Ignition SCADA system by using KEPServerEX OPC-UA Server (Monitor and Control).
5. Evaluate the performance of the OPC-UA communication under different hardware specification and security profile configuration using UaExpert OPC-UA client.

CHAPTER 2

LITERATURE REVIEW

2.1 Theory

2.1.1 Data Communication

Data communications are the process of sending and receiving digital data or information between two locations or points over a data network [8]. Communications only occur when there is a path or medium either wired or wireless for the data to transfer [8].

2.1.2 Network Topologies

Network topologies are the arrangement of nodes or links in a data communication network. A few basics type of network topologies is shown in figure 2.1.

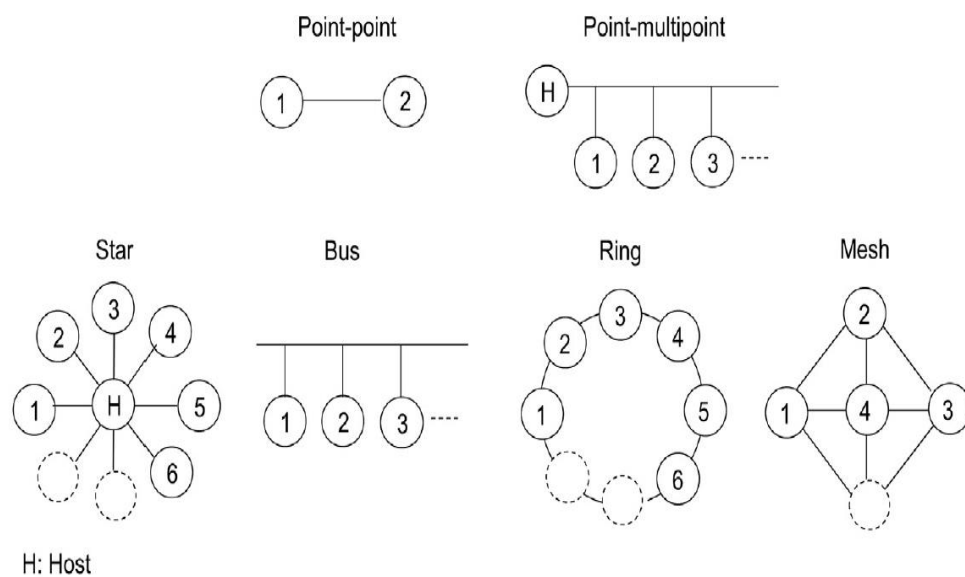


Figure 2.1 Network Topologies [8]

2.1.3 Interprocess Connection

Interprocess communication is the processes or applications that sharing data among themselves whether within a device or many devices in a network [8]. Client/server architecture is the most commonly used architecture among others that used to establish data transfer between two partners [8].

2.1.4 OPC (Open Platform Communication)

OPC (Open Platform Communications) or originally OLE (Object Linking and Bedding) for process control is a technology and an open standard for interprocess communication. There are two main components of the OPC which are the OPC Server and the OPC Client. An OPC Server is a software application that works as a “standardized” driver that read and write any data sources such as PLCs, RTUs, and etc [9]. Where an OPC Client is a software that written to communicate with the OPC Server, to send or received data or information from the OPC Server [9]. Figure 2.2 shows the overviews of the OPC family.

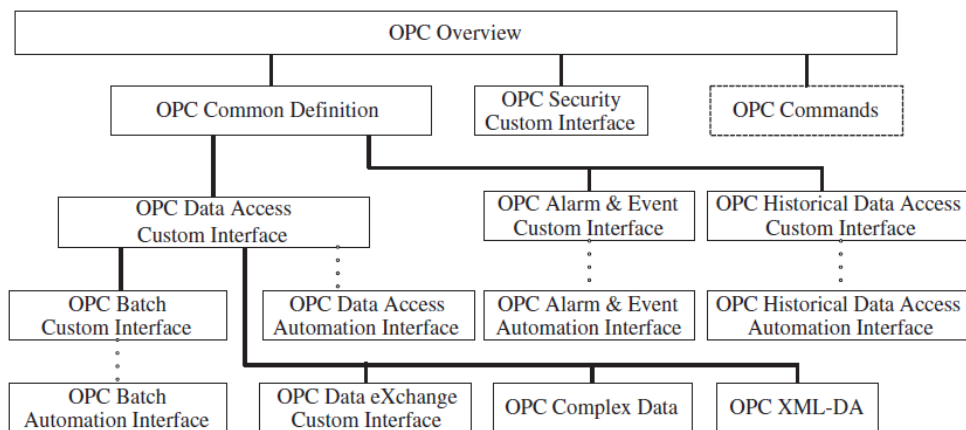


Figure 2.2 Overview of OPC Family [10]

The specification of OPC Classic can be categories into three main definitions; DA (Data Access), AE (Alarm & Events), and HDA (Historical Data Access). The OPC DA defines the exchange of data (values, time and quality information), where the OPC AE defines the exchange of the alarm and event type information, variable states and state management, and the last is OPC HDA. In OPC HAD, the specification defines the query methods and analytics that may be applied to the historical, time-stamp data.

2.1.5 OPC-UA (OPC Unified Architecture)

OPC-UA can be considered as the next generation of the OPC Classic. According to [11], there are many new functions are added in the OPC-UA. In [11], authors claim that OPC-UA is platform independent, secure, and reliable technologies compared to the Class OPC. OPC-UA provides a consistent, integrated Address Space and service model. Besides that, OPC-UA also allows data to be exposed in many diverse formats, adds support for many relations between nodes and designed for a wide range of servers. Figure 2.3 shows the illustrator of major elements of the server/client architecture of OPC-UA:

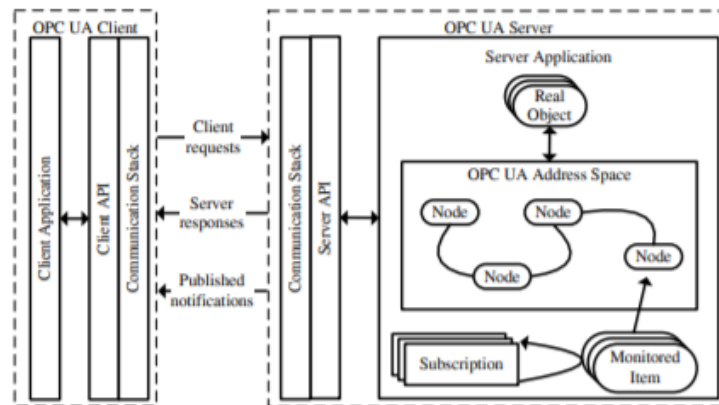


Figure 2.3 OPC-UA Server/Clients Architecture [12]

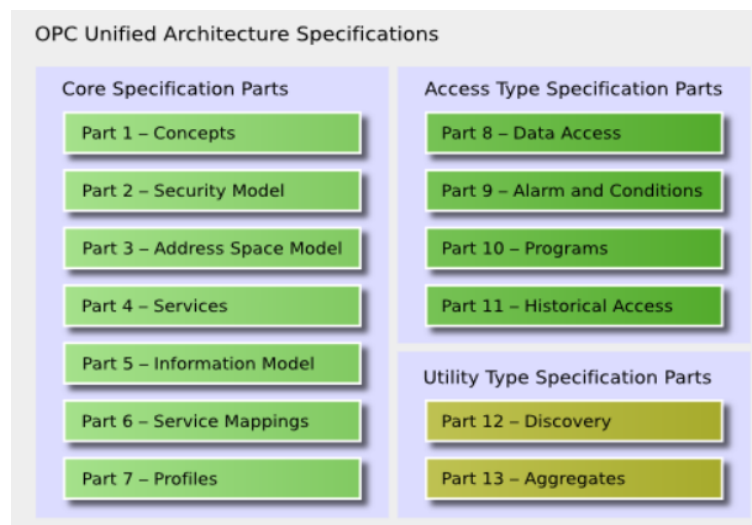


Figure 2.4 OPC-UA Specification [13]