Benchmarking of existing OPC UA implementations for Industrie 4.0-compliant digitalization solutions

Hermann Haskamp, Michael Meyer, Romina Möllmann, Florian Orth and Armando Walter Colombo

Institute for Industrial Informatics, Automation and Robotics (I2AR)
Department of Technology
University of Applied Sciences Emden/Leer
Emden, Germany
Email: {hhaska, mimeye, rmoell, forth}@technik-emden.de, awcolombo@ieee.org

***Abstract*—A standardized Communication / Information Protocol is necessary for implementing the 4 upper digitalization layers of Industrie 4.0-compliant solutions. OPC UA is one of the leading and preferred technology. Currently, there are many different OPC UA implementations available on the market. This paper gives an overview of existing solutions and shows the results of an initial benchmarking based on a set of evaluation criteria defined by the authors. Moreover, the manuscript identifies both free-of-charge and commercial solutions that are more adequate for implementing industrial applications.**

Keywords—OPC UA, Industrie 4.0, Industrial Digitalization, Benchmarking

I. INTRODUCTION

In order to support the digitalization of industrial systems, the DIN specification 91345 (Reference Architecture Model for Industrie 4.0 (RAMI4.0)) propose the design, development and implementation of a 6-layer SW/HW stack. [1] Particularly the communication and information (3rd and 4th) layers are responsible for guaranteeing connectivity and interoperability of digitalized assets.

Among other communication and information technologies and associated data and information models, OPC UA is one of the preferred for implementing Industrie 4.0-compliant solutions. For developers but also users, inclusive researchers and practitioners, the market offers a broad pallet of commercial and cost-free OPC UA solutions. The selection of the most adequate OPC UA implementation for being used in a given industrial solution requires the evaluation of many technical and organizational factors.

In this paper, the authors present an overview of existing OPC UA implementations, classified into the two categories addressed before. After defining a set of 15 evaluation criteria for the one or the other category, an initial benchmarking of the selected solutions, in front of those criteria is summarized and used for generating a very first set of recommendations for the selection.

After a brief introduction to the OPC UA basics in Section II, Section III describes the set of evaluation criteria that are used in Section IV for performing the benchmarking. Section V presents a first initial analysis of the benchmarking. Section VI rounds up the paper identifying major issues discovered in the benchmarking and outlooks future extensions of the work.

II. OPC UA BASICS

Since 2007, the OPC Foundation works on a M2M communication protocol called OPC UA (https://opcfoundation.org/). This protocol is already an IEC standard (IEC 62541) and currently defined by 13 specifications. These specifications are assigned to the parts Core Specifications, Access Specifications and Utility Specifications as shown in Figure 1. One of the major purposes of the standard is the definition of a generic data model with nodes and references which satisfies all necessary requirements for modelling data in a digitalized industrial environment. Therefore, the OPC Foundation works in close cooperation with users, researchers, practitioners and industrialists, so that the specifications contain technologies, basic concepts and methods for communication. [2]

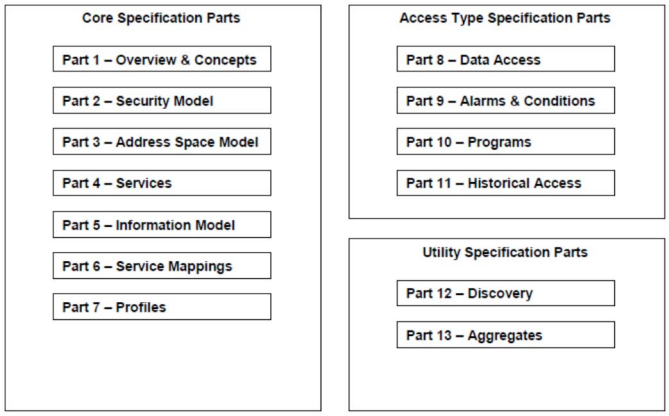

Figure 1: OPC UA Specifications [3]

One of the major advantages is that OPC UA enables interoperability between different components and systems (assets) in an Industrie 4.0-compliant environment. For that reason, it is usable for specifying and implementing the communication and information layers between those assets, e.g. products, mechatronic components from different manufacturers. Furthermore, OPC UA uses established standards for data

transport and security, which is an important requirement for being used by the industry.

Since digitalized assets are an essential part of industrial enterprise architectures, the OPC Foundation collaborates with other organizations to specify Companion Standards such as ISA 95 / IEC 62264 (https://isa-95.com) and AutomationML (https://www.automationml.org). One important part of these collaborations is the definition of methods and rules for mapping existing information models to OPC UA. [2]

III. Evaluation Criteria for Benchmarking OPC UA Implementations

In order to proceed with a credible industrial-oriented benchmarking of OPC UA implementations, a set of evaluation criteria needs to be defined. The explanation of the criteria with the corresponding sub-criteria defined by the authors, as well as the reasons for selection are given in the following section.

- **Version:** The version number indicates the state of the software implementation used in the benchmarking.
- **Actuality:** The last change date of the software implementation is recorded in addition to the version number to be able to conclude the actuality.
- **Developer:** The following classification of the developers is applied:
    - Private Person / Community
    - Research Institute / University
    - OPC Foundation
    - Company
- **Language:** The programming language is listed to indicate, among others, the required programming skills.
- **License:** It is important for the user to know the IPR rules (specifically rights of use and copyrights). Commercial Software has company-specific licenses which are not comparable. For this reason, this criterion is only used for free software products.
- **Demo:** To be able to test commercial software before purchasing, some companies offer demo versions. These are limited in time and / or have a smaller range of functions. This category is useless for free software and therefore it is only applied to commercial software.
- **Certificate:** As mentioned above, OPC UA is described by 13 specifications. The OPC Foundation checks implementations for compliance with the profiles in the 7th part of the OPC UA specification and awards them a certificate. Servers or clients do not have to fulfill all profiles to receive a certificate. In this category, it is not distinguished which profile is fulfilled. [4]
    - Yes: There is a certificate for this software.
    - No: There is no certificate for the software.
- **Type**: To show the abstraction level of the software, the Type category is selected. The OPC UA stack offers the base for the implementation of OPC UA, but further programming is required to develop a server or client. Software has a server and client functionality, as soon as the software offers methods or functions for the realization of these two. Therefore, the user does not need to mind the stack level.
    - Stack: basic functionality
    - Client: one-sided extended functionality
    - Server: one-sided extended functionality
    - Client + Server: extended functionality on both sides
- **Mappings**: OPC UA offers the possibility to choose between different protocols for encoding, data transport and security. On the one hand there are two types of encoding: UA Binary and XML. On the other hand there are several transport protocols: UA TCP and HTTPS. The OPC Foundation divides the combination of the encoding and the transport protocol into the following three facets [5]:
    - UA-TCP UA-SC UA Binary: Binary encoded and TCP transported.
    - HTTPS UA Binary: Binary encoded and HTTPS transported
    - HTTPS UA XML: XML encoded and HTTPS transported
- **Services**: According to the first OPC UA specification (see [3]) a service is a client-callable operation in an OPC UA server. Services therefore represent the functional possibilities of the server and client, respectively (layer 5 and 6 of the RAMI4.0 digitalization dimension [6]). An abstract definition of the intended services is given in the fourth part of the OPC UA specification [7]. The specification organizes them in 10 different service sets.

  To get a functional comparison of the OPC UA implementations, for this benchmarking, a set of basic services have been considered:
    - Discovery Service Set: GetEndpoints, FindServers
    - NodeManagement Service Set: AddNodes, AddReferences, DeleteNodes, DeleteReferences
    - View Service Set: Browse
    - Attribute Service Set: Read, HistoryRead, Write
    - Method Service Set: Methods
    - MonitoredItem Service Set: CreateMonitoredItems
    - Subscription Service Set: CreateSubscription, Publish
- **Security Policies**: Part 2 of the OPC UA specification [8] describes the security model including different security mechanisms and parameters. Before opening a Secure Channel, server and client have to agree upon the security mechanism used for this communication session. Security Policies can be used by the server to announce the supported security mechanisms. Part 7 of the OPC UA specification [4] defines four different SecurityPolicies as Profiles:
    - None: No security configurations.
    - Basic128Rsa15: Medium security configurations.
    - Basic256: Medium to high security configurations.
    - Basic256Sha256: High security configurations.
- **User Token**: Authentication is an important part of the OPC UA security model as described in the second part of

the specifications [8]. Part 7 [4] defines the following ways to exchange user identity tokens:

  - Anonymous: No user identity required.
  - User Name Password: Combinations of Names and Passwords for user identity.
  - X.509 Certificate: Public / private key pair for user identity.
  - Issued Token: Kerberos tokens for user identity.
  - Issued Token Windows: Windows implementation of Kerberos tokens for user identity.

- **Documentation**: The quality and quantity of supporting documentation for the source code is very important for developers and users, particularly helpful when starting to develop and/or use own OPC UA server and clients. The existence of documented use-cases, application examples and tutorials makes a great difference for the user of an OPC UA implementation. Therefore, existing documentation has to be adequately ranked, for example as follows:
  - Very Few: Source code with (short) documentation
  - Few: Commented source code, (short) documentation and some examples
  - Much: Commented source code, extensive documentation and some examples or tutorials
  - Very Much: Commented source code, extensive documentation, examples and tutorials
- **Operating system**: Target platforms.
- **TRL**: Technology Readiness Level as defined by the European Commission [9]. The TRL serves as an indicator for the development state of the implementation. The values range from 1 to 9, 1 being only a basic principle and 9 being a complete and thoroughly tested system.

IV. BENCHMARKING

A. Method

As a first step, a screening on existing OPC UA implementations was necessary and allows a first classification of free-of-charge and commercial solutions. Remark: the authors are aware that there might be solutions which are missed.

With the exception for the TRL, the evaluation of the criteria described in chapter III is based on documentations, online information and the source code (if available). Furthermore, only functionalities that are fully implemented and not just in development state have been considered. For evaluating the TRL, the respective developer or the publisher were directly contacted by the authors, in order to obtain information on the environment in which the implementation is currently applied.

B. Results

The results of the screening is presented in two separated parts. The first one consists of the free-of-charge OPC UA implementations and the second one consists of commercial implementations. Note: The reason for this separation is the application of different evaluation criteria for both kind of screened solutions.

Name		**ASNeG**[1]	**Eclipse Milo**[2]	**FreeOpcUa C++** [3]
Version		1.0.8	-	-
Actuality		03.2016	02.2017	03.2017
Developer		Company	Community	Community
Language		C/C++	Java	C++
License		Apache 2	Eclipse Public 1.0	LGPL
Certificate		No	No	No
Type		Client + Server	Client + Server	Client + Server
Mappings	UA-TCP UA-SC UA Binary	x	x	x
	HTTPS UA Binary	x	x	
	HTTPS UA XML	x	x	
Services	GetEndpoints		x	x
	FindServer		x	x
	AddNodes		x	x
	AddReferences		x	x
	DeleteNodes		x	
	DeleteReferences		x	
	Browse		x	x
	Read	x	x	x
	HistoryRead		x	
	Write	x	x	x
	Methods		x	
	CreateMonitoredItems	x	x	x
	CreateSubscription		x	x
	Publish		x	x
Security Policies	None		x	x
	Basic128Rsa15		x	
	Basic256		x	
	Basic256Sha256		x	
User Token	Anonymous	x	x	x
	User Name Password		x	
	X.509 Certificate		x	
Documentation		few	very much	few
Operating system		Windows, Linux		Windows, Linux
TRL				4

Table 1: free OPC UA implementations

[1] Source: http://www.asneg.de/ [Accessed 15 03 2017].
[2] Source: https://github.com/eclipse/milo [Accessed 15 03 2017].
[3] Source: http://freeopcua.github.io/ [Accessed 15 03 2017].

Name		FreeOpcUa Python[4]	node-opc-ua[5]	OPC UA .NET[6]	OPCUA4j[7]	OPC UA ANSIC[8]	OPC UA Client[9]	open62541[10]	OPyCua[11]	UAF[12]
Version		0.90.2	0.0.61	1.03.341	0.9.5	1.03.340	1.5.2	0.2-rc2	prealpha	-
Actuality		03.2017	02.2017	02.2017	03.2015	03.2017	02.2017	12.2016	06.2012	09.2016
Developer		Community	Community	OPC Foundation	University	OPC Foundation	Community	Research Institute	Private Person	University
Language		Python	JavaScript	C#	Java	C	C#	C99	Python	C++, Python
License		LGPL	MIT	RCL, GPL 2.0, MIT	CC 3.0 BY-SA	RCL, GPL 2.0, MIT	MIT	MPLv2.0	GPL v3	GNU
Certificate		No	No	Yes	No	No	No	No	No	No
Type		Client + Server	Client + Server	Stack	Server	Stack	Client	Client + Server	Client + Server	Client
Mappings	UA-TCP UA-SC UA Binary	x	x	x	x	x	x	x	x	x
	HTTPS UA Binary			x		x	x			
	HTTPS UA XML			x						
Services	GetEndpoints	x	x	x		x	x	x		x
	FindServer	x	x	x		x	x	x		x
	AddNodes	x		x	x	x	x	x		x
	AddReferences	x		x	x	x	x	x		x
	DeleteNodes	x		x		x	x	x		x
	DeleteReferences	x		x		x	x	x		x
	Browse	x	x	x	x	x	x	x		x
	Read	x	x	x	x	x	x	x		x
	HistoryRead	x		x		x	x			
	Write	x	x	x		x	x	x		x
	Methods	x	x	x		x	x	x		x
	CreateMonitoredItems	x	x	x	x	x	x	x		x
	CreateSubscription	x	x	x	x	x	x	x		x
	Publish	x	x	x	x	x	x	x		x
Security Policies	None	x	x	x	x	x	x	x	x	x
	Basic128Rsa15	x	x	x	x	x	x			
	Basic256	x	x	x	x	x	x			
	Basic256Sha256		x	x		x	x			
User Token	Anonymous	x	x	x	x	x	x	x	x	x
	User Name Password	x	x	x	x	x	x	x		x
	X.509 Certificate	x		x		x	x			
Documentation		much	much	few	very few	very few	very few	very much	few	few
Operating system		Windows, Linux	Windows, Linux, MacOS	Windows	Windows, Linux	Windows, Linux	Windows	Windows, Linux, QNX, Android	Windows, Linux	Windows, Linux
TRL							7	7		

Table 2: free OPC UA implementations

[4] Source: https://github.com/freeopcua/python-opcua [Accessed 15 03 2017].
[5] Source: http://node-opcua.github.io/ [Accessed 15 03 2017].
[6] Source: http://opcfoundation.github.io/UA-.NET/ [Accessed 15 03 2017].
[7] Source: https://code.google.com/p/opcua4j/ [Accessed 15 03 2017].
[8] Source: http://opcfoundation.github.io/UA-AnsiC/ [Accessed 15 03 2017].
[9] Source: https://github.com/convertersystems/opc-ua-client [Accessed 15 03 2017].
[10] Source: http://open62541.org [Accessed 15 03 2017].
[11] Source: https://sourceforge.net/projects/opycua/?source=navbar [Accessed 15 03 2017].
[12] Source: https://github.com/uaf/uaf [Accessed 15 03 2017].

Name		OpenOpcUA[13]	Matrikon OPC UA SDK [14]	proSys[15]	Softing OPC Development Tool C++ [16]	Softing OPC Development Tool .NET [17]	UA .NET OPC UA SDK [18]	UA ANSI C OPC UA SDK[19]	UA C++ OPC UA SDK[20]	UA High Perfor. OPC UA SDK[21]
Version		1.0.4.4	-	2.0	5.52.0	1.40.0	2.5.3	1.7.0	1.5.4	1.0.0
Actuality		10.2016	03.2017	11.16	09.2016	09.2016	12.2016	07.2016	12.2016	10.2016
Developer		Company	Company	Company	Company	Company	Company	Company	Company	Company
Language		C, C++	C++	Java, C, C++, .Net	C++	C#	C#	ANSI C	C++	C99
Demo		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Certificate		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Type		Client + Server	Client + Server	Client + Server	Client + Server	Client + Server	Client + Server	Client + Server	Client + Server	Server
Mappings	UA-TCP UA-SC UA Binary	x	x	x	x	x	x	x	x	x
	HTTPS UA Binary	x		x	x	x	x		x	
	HTTPS UA XML					x			x	
Services	GetEndpoints	x	x	x	x	x	x	x	x	x
	FindServer	x	x	x	x	x	x	x	x	x
	AddNodes		x	x	x	x	x	x	x	x
	AddReferences		x	x	x	x	x	x	x	x
	DeleteNodes		x	x	x	x	x	x	x	x
	DeleteReferences		x	x	x	x	x	x	x	x
	Browse	x	x	x	x	x	x	x	x	x
	Read	x	x	x	x	x	x	x	x	x
	HistoryRead	x	x	x	x	x	x	x	x	
	Write	x	x	x	x	x	x	x	x	x
	Methods	x	x	x	x	x	x	x	x	x
	CreateMonitoredItems	x		x	x	x	x	x	x	x
	CreateSubscription	x	x	x	x	x	x	x	x	x
	Publish	x	x	x	x	x	x	x	x	x
Security Policies	None	x	x	x	x	x	x	x	x	x
	Basic128Rsa15	x	x	x	x	x	x	x	x	
	Basic256	x	x	x	x	x		x	x	
	Basic256Sha256		x	x				x	x	
User Token	Anonymous	x	x	x	x	x	x	x	x	x
	User Name Password	x	x	x	x	x	x	x	x	x
	X.509 Certificate		x	x	x	x	x	x	x	x
Documentation		very much	very much	much	very much	very much	very much	very much	very much	very much
Operating system		Windows, Linux, VxWorks	Windows, Linux	Windows, Linux	Windows, Linux, VxWorks	Windows	Windows	Windows, Linux, VxWorks, QNX, EUROS	Windows, Linux, VxWorks, QNX	Windows, Linux
TRL		9		9	9	9				

Table 3: commercial OPC UA implementations

[13] Source: http://www.openopcua.org/ [Accessed 15 03 2017].
[14] Source: http://www.matrikonopc.de/ [Accessed 15 03 2017].
[15] Source: https://www.prosysopc.com/products/ [Accessed 15 03 2017].
[16] Source: http://industrial.softing.com/de/produkte/software/opc-development-toolkits.html [Accessed 15 03 2017].
[17] Source: http://industrial.softing.com/de/produkte/software/opc-development-toolkits.html [Accessed 15 03 2017].
[18] Source: https://www.unified-automation.com/products/server-sdk/net-ua-server-sdk.html [Accessed 15 03 2017].
[19] Source: https://www.unified-automation.com/products/server-sdk/ansi-c-ua-server-sdk.html [Accessed 15 03 2017].
[20] Source: https://www.unified-automation.com/products/server-sdk/c-ua-server-sdk.html [Accessed 15 03 2017].
[21] Source: https://www.unified-automation.com/products/server-sdk/highperf-ua-server-sdk.html [Accessed 15 03 2017].

V. A FIRST EVALUATION

The tables above show that there are several OPC UA implementations both free-of-charge and with costs that differ in the implemented functionalities. As intended by the OPC Foundation, different programming languages and operating systems are covered. This is a contrast to the "classic" OPC DA which depends on the Microsoft DCOM communication technology and therefore only runs on Windows systems.

Tables 1 and 2 show the free-of-charge OPC UA implementations. There are many differences regarding the realized functionalities listed as criteria in the benchmarking. This is a result of the varying development state of the evaluated solutions. All these implementations are Open Source projects, the majority of them offered as repositories on the development platform GitHub. Therefore, they are subject to various and continuous changes. However there are some projects like "OPyCua" that have not been updated for quite some time. Besides those outdated projects all free-of-charge implementations have covered many of the listed Services and other functionalities. Two of the implementations, "Eclipse Milo" and "OPC UA .NET", even comply with all the examined criteria.

Table 3 shows the commercial OPC UA implementations. With the exception of only one solution, all others are certified by the OPC Foundation. For every SDK the respective companies offers a demo version, mostly a time limited version with full functionality. The differences between the implementations are rather in the criteria Mappings, Security Policies and User Token than in Services. There is only one implementation that complies with all the examined criteria of the benchmarking, it is the "UA C++ OPC UA SDK" from Unified Automation.

In relation to the criterion "Mapping", it is important to recall that only a few of the compared implementations support the HTTPS UA XML Mapping. The majority concentrates on the UA-TCP UA-SC UA Binary Mapping. For that requirement our comparison shows only 5 free-of-charge as well as commercial implementations. Generally, the implementations are not precisely described, although the Specification Part 7 describes Profiles and Facets for servers, clients or security. Moreover, developer and publisher use no common description for their implementations.

The Documentation is a difficult criterion for being correctly evaluated since there are many differences not only in quantity but also in the quality of the offered documentation.

A noticeable fact respecting the certification is that the OPC Foundation clearly defines that SDKs and other tools in contrast to servers and clients cannot be certified. [10] However, the Foundation lists various SDKs as certified products on their webpage.

Regarding the TRL only some developers or companies were able to position their solutions in the TRL scale.

VI. CONCLUSION

OPC UA is being stablished as one of the recommendable communication technology for implementing communication, information, functional and business layers of Industrie 4.0-compliant solutions. In this paper, the authors present the results of a benchmarking of OPC UA solutions existing in the market.

After an intensive analysis of the market, solutions have been classified into two groups, those that are offered free-of-charge and those that are commercial.

In order to perform the benchmarking, 15 major evaluation criteria were initially defined and used to qualitatively and quantitatively compare 21 OPC UA implementations, 12 in the free-of-charge and 9 in the commercial solution categories.

On the one hand, the benchmarking has shown that there is only one commercial implementation that complies with all the evaluation criteria defined by the authors: the "UA C++ OPC UA SDK" developed by Unified Automation. On the other hand, two of the free-of-charge evaluated implementations were able to fulfill the criteria for the category of solutions: "Eclipse Milo" and "OPC UA .NET".

In general, it is possible to detect a trend showing the increasing offer of free-of-charge implementations that are well documented and ready for being also applied in industrial environments.

Future extensions of this work will be focusing, among others, on (i) extending the quality and quantity of benchmarking criteria according to the increasing number of requirements for Industrie 4.0-compliant solutions, (ii) extending the number of solutions to be benchmarked in both categories (depending on the offer in the market).

REFERENCES

[1] ZVEI; VDI, "Statusreport: Referenzarchitekturmodell Industrie 4.0 (RAMI4.0)," 2015.

[2] OPC Foundation, "opcfoundation," [Online]. Available: https://opcfoundation.org/wp-content/uploads/2014/03/OPC_UA_I_4.0_Pioneer_US_v2.pdf. [Accessed 15 03 2017].

[3] OPC Foundation, "OPC UA Part 1 - Overview and Concepts 1.03 Specification.pdf," 2015.

[4] OPC Foundation, "OPC UA Part 7 - Profiles 1.03 Specification.pdf," 2015.

[5] OPC Foundation, "OPC UA Part 6 - Mappings 1.03 Specification.pdf," 2015.

[6] DIN, *DIN SPEC 91345,* 2016.

[7] OPC Foundation, "OPC UA Part 4 - Services 1.03 Specification.pdf," 2015.

[8] OPC Foundation, "OPC UA Part 2 - Security Model 1.03 Specification.pdf," 2015.

[9] European Commission, "HORIZON 2020," 2014. [Online]. Available: http://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf. [Accessed 12 03 2017].

[10] OPC Foundation, "Certification," [Online]. Available: https://opcfoundation.org/certification/overview-benefits/. [Accessed 06 06 2017].