Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

SANTERI TURTA

OPC UA Information modelling for automation system interfaces

DEGREE PROGRAMME IN ELECTRICAL AND AUTOMATION ENGINEERING
2025

ABSTRACT

Turta, Santeri: OPC UA Information modelling for automation system interfaces
Bachelor's thesis
Electrical and Automation Engineering
January 2026
Number of pages: 31

The objective of this thesis was to design and implement an OPC UA information model for Cimcorp MBR gantry robot systems to describe and standardize automation system interfaces. The aim was to create a structured, human-readable representation of the PLC interface that would simplify development, improve interoperability, and reduce engineering overhead.

The thesis was conducted as a design and implementation project in collaboration with my employer, Cimcorp Oy. Theoretical background was gathered from the OPC UA standards and related literature to establish an understanding of information modelling principles. The model was designed following the OPC UA specifications and best practices using UaModeler software. A requirement analysis was performed, and an iterative approach was applied to design, validate, and refine the model. The custom information mode was implemented in its own OPC UA namespace with a modular hierarchy reflecting the system architecture. The model was validated against the OPC UA standard using built-in validation tools and reviewed with the Cimcorp´ device development team.

The results showed that a custom OPC UA information model can effectively describe and standardize automation system interfaces. The created model provides a unified structure for the development of HMI and facilitates integration between product variants. Although real-world deployment was not completed within the thesis schedule, the model was prepared for implementation in upcoming projects.

It was conducted that OPC UA information modelling improves the maintainability and scalability of automation systems by enabling structured and standardized data representation. The approach adopted in this work can be generalised to other automation systems, demonstrating the potential of OPC UA information models as a future-proof solution for unified industrial communication.

Keywords: OPC UA, information modelling, automation systems, interface design

PREFACE

This thesis was conducted as part of the Bachelor of electrical engineering degree at Satakunta University of Applied Sciences, in collaboration with Cimcorp Oy. The work explores the possibilities of describing an automation system interface using OPC UA information modelling.

I would like to express my sincere gratitude to my thesis supervisors Jani Tuomola and Ari Stjerna, for valuable guidance and feedback during the thesis. I also want to thank my colleagues at Cimcorp, particularly the Device Development team, for their practical suggestions and insights during the development phase of the information model´. Their experience in automation systems was essential to the success of this work. Finally, I extend my gratitude to Satakunta University of Applied Sciences for providing the academic environment that allowed me to write this thesis.

CONTENTS

LIST OF SYMBOLS AND TERMS

OPC UA	Open platform communications unified architecture, a communication standard
OPC Classic	OLE for Process Control Classic, a communication standard
PLC	Programmable logic controller
HMI application	Human-Machine interface application, an application to interact with and control industrial machinery
UI	User interface
MBR gantry robot system	Cimcorp´s portal robot product
UaModeler	OPC UA Information modelling software
GUI application	Graphical user interface application, a graphical application to interact with a computer or a device
SDK	Software development kit
COM	Component object model, a software architecture by Microsoft
DCOM	Distributed component object model, a software architecture by Microsoft

1 INTRODUCTION

In modern industrial environments, the complexity of automation systems has increased significantly, leading to challenges in interface design, data exchange and system integration. Traditional communication protocols and standards often result in fragmented interfaces and high engineering overhead. OPC UA is a communication standard most used in industrial automation that provides solutions to many of the problems of communication in modern automation systems.

Despite the advantages of OPC UA, many automation systems still rely on unstructured or proprietary interfaces that complicate development and maintenance. HMI applications often suffer from inconsistent data structures and lack of context for the data which leads to reduced system flexibility. Cimcorp recently adopted a new generation of PLC´s that utilize OPC UA. With the new generation of PLC´s it was decided to also develop a new web-based UI. OPC UA was entirely new to Cimcorp and a proper description of the OPC UA interface didn´t exist which increased development time. Motivated by the need to describe the automation system interfaces, this thesis explores the potential of OPC UA information modelling.

This thesis aims to design and implement a custom OPC UA information model tailored for Cimcorp MBR gantry robot systems. The model is designed to act as a human readable interface description, but the model aims to achieve this by providing a structured interface for HMI development, standardizing data representation across product variants

The scope of this thesis is limited to the development and evaluation of an OPC UA information model for the MBR gantry robot system. Modelling the physical components is excluded from the information model since solving the problem of describing the interface is prioritized.

The information model was developed and validated using UaModeler.

2 THE HISTORY AND FUNDAMENTALS OF OPC CLASSIC AND OPC UA

2.1 Evolution of OPC Classic

2.1.1 Rationale Behind the Development of OPC Classic

In the beginning of the 1990s the amount of software-based automation systems increased rapidly. This development caused a challenge; many of the devices used in the automation systems required their own drivers to enable communication with the Windows-operating system. Each device in the system required its own solution so device specific drivers, communication protocols and interfaces were being developed uncontrollably. This resulted in the automation systems becoming complex and hard to maintain. (Industry40tv, 2020)

This problem was solved by numerous automation companies that joined forces and developed a standardized way for devices and systems to communicate and transfer data. As a result, OPC (OLE for Process Control), nowadays known as OPC Classic was born. The rationale behind OPC Classic was to enable automation devices to communicate through a singular standardized interface. This way each device didn’t have to communicate with E.g. their specific controller and instead the devices were able to exchange information with a centralized OPC server. (Industry40tv, 2020)

2.1.2 Operational Principles of OPC Classic

The way Classic OPC works is that different client applications (E.g. control room application) can use different automation devices such as sensors through OPC servers. In this system the OPC server works as an intermediary; it understands the structure and content of the data created by the different devices and relays the data to client applications. This way the client application only needs to know how to communicate with the OPC server and what data it needs instead of understanding and adapting to each different device. (Real Time Automation, n.d.)

The communication of Classic OPC is based on Microsoft COM (Component Object Model) and DCOM (Distributed Component Object Model) protocols which enabled transmission of data and commands between programs. At the time this was an innovative solution, however it was also dependent on the Windows operating system and faces challenges related to security and scalability. (Real Time Automation, n.d.)

2.2 Fundamentals of OPC UA

OPC Unified Architecture (OPC UA) is a platform independent service-based architecture that combines all the features of the OPC Classic specifications into a single expandable standard. The layered approach of OPC UA meets all the original design requirements such as functional equivalence with OPC Classic, platform independence, security, scalability and comprehensive information modelling. Due to these properties, OPC UA can offer a structure to handle and define complex information in varying production environments. (OPC Foundation, n.d.)

OPC UA offers comprehensive information security features that meet the requirements of the modern industry. OPC UA supports encrypting messages on different levels, signatures to confirm the origin of the packet and packet sequencing which counters replay attacks. Additionally, X509-certificates recognize client, and server programs and restricts access to only authorized systems. Users can login with E.g. credentials, authenticators or domains and user permissions can be controlled via views in the address space. All actions can be audited with logging. (OPC Foundation, n.d.)

OPC UA´s information modelling framework enables defining and extending complicated data structures in an object-oriented way. It´s a crucial part of OPC UA architecture and it enables defining information related to industrial systems and creating and maintaining the semantic relationships within the system. OPC Foundation offers many ready to use core information models for many different industrial sectors however, organizations can extend these models to meet their specific requirements. The information modelling process involves searching for information, reading and writing data and informing client applications about events and changes in data. (OPC Foundation, n.d.)

2.3 Communication Models in OPC UA

OPC UA standard supports the traditional client/server model where a client requests specific information and the server responds to the request. OPC UA models clients and servers as interacting partners and every system can contain multiple clients and servers. Additionally, each client can simultaneously communicate with a single server or multiple servers and servers can communicate with a single client or multiple clients. (OPC Foundation, 2024a)

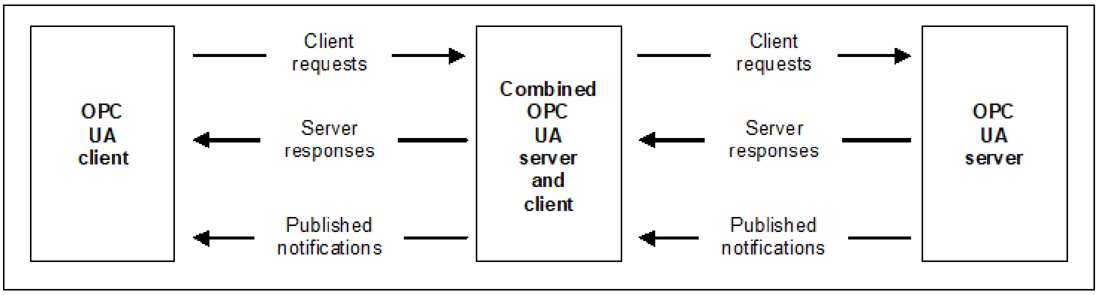

Image 1. OPC UA System architecture (OPC Foundation, 2024a)

An OPC UA application also has the possibility to combine components from clients and servers. This makes it possible for an application to act as both the server and the client simultaneously (image 1). (OPC Foundation 2024a).

Another traditional communication model is the so called PubSub-model where publishers send information without knowing who receives it and subscribers listen to specific queues without direct contact to the publishers. This communication model specifically suits real-time data transfer and scalable IoT and Industry 4.0 applications. (Prosys OPC Ltd, 2021)

In addition to the previously mentioned communication models there have been developed hybrid models and extensions which better support the real-time communication, field level exchange and specific requirements of industrial systems.

3 OPC UA INFORMATION MODELLING

3.1 The OPC UA Address Space

3.1.1 Structure of the Address Space

OPC UA combines information, alarms, events and historical information as a single unified address space that provides clients with access to all necessary information through a uniform service model. Unlike traditional hierarchies OPC UA supports complex references between nodes so data can be displayed in different views according to the client's needs. This makes OPC UA suitable for various use cases. (OPC Foundation, 2024a)

The address space presents and organizes the information in a hierarchical structure (image 2). OPC UA address space consists of different types of nodes. OPC UA address space defines different base types for nodes such as object node, variable node, object type node and variable type node. Each node belongs to a specific group of nodes and can be identified with a unique node-ID. A group of nodes can always be identified with a unique namespace URI. (Profanter, 2023)

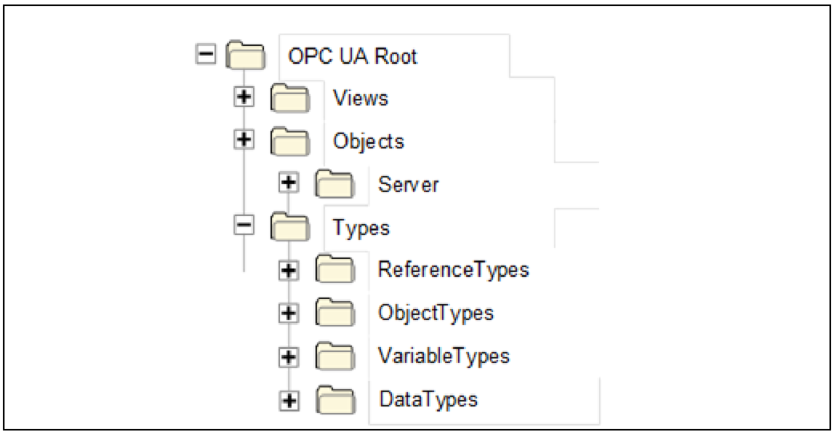

Image 2. Hierarchy of OPC UA address space (OPC Foundation, 2024b)

Relationships between nodes are defined with references. A node that contains a reference is a source node and the referenced node is a target node.

(OPC Foundation, 2024b) All references are defined using reference types and they can be directional (OPC Foundation, 2024a).

3.1.2 The Impact of the Address Space on Information Modelling

OPC UA address space defines the OPC UA metamodel which offers a theoretical structure for the base information model. The base information model creates a base for creating standardized or domain specific information models that are tailored to a specific industry. The address space defines the node classes, base types and restrictions. The information model uses the address space definitions to define its own types, restrictions and carefully defined instances. In the end the server's concrete data is created based on the information model. (Tulka, 2022)

3.2 Fundamentals of Information Modelling

3.2.1 What Is an OPC UA Information Model and What Is It Used For

OPC UA information models are based on data modelling. The purpose of data modelling is to define the required rules and building blocks to display the information in the OPC UA address space. OPC UA takes advantage of many principles from object-oriented programming such as inheritance and type hierarchy. The purpose of strong typing is to make sure that clients can handle all instances of a type in the same way. Due to the type hierarchy, clients can deal with base types and ignore more specific extensions if needed. (Tulka, 2022)

In essence, an OPC UA information model is a collection of nodes and references between different nodes. The information model's function is to define how the nodes and data structures are placed in the server's address space. (Profanter, 2023)

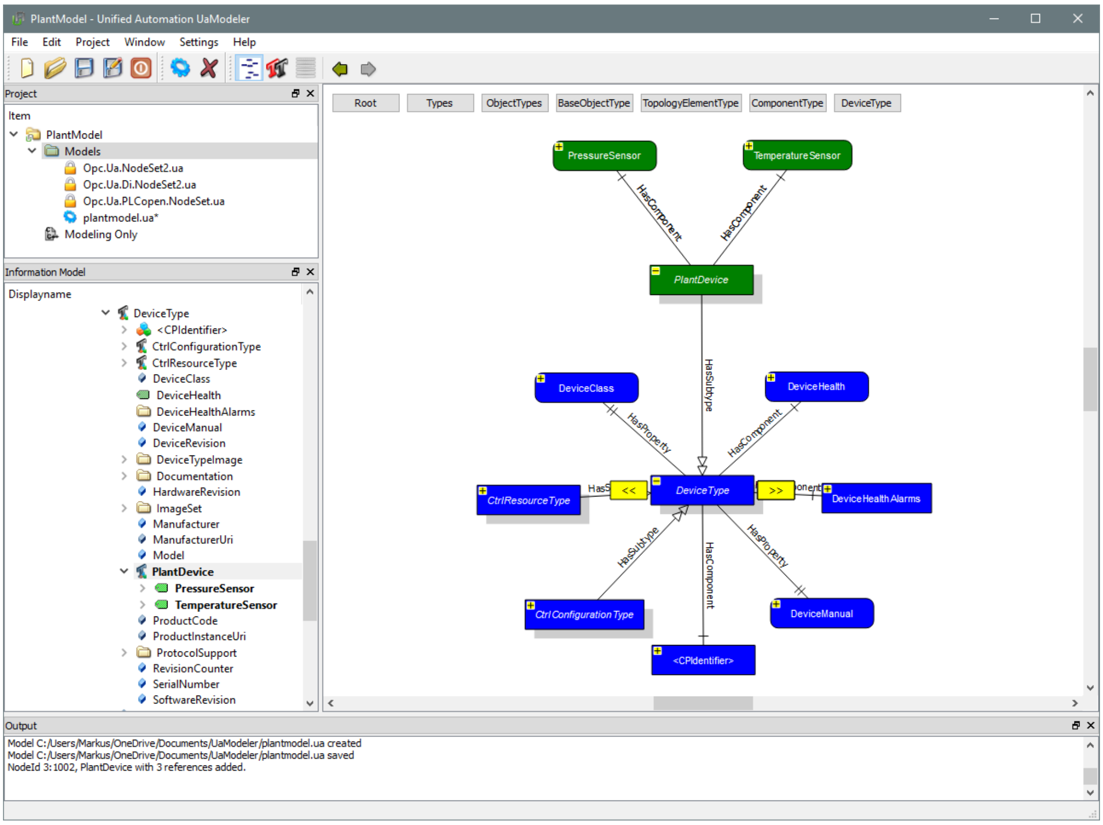

Image 3. Information model in UaModeler application (Prosys OPC Ltd, n.d.)

In addition to utilizing OPC UA information models in handling complicated information structures and making sure systems are compatible with each other the OPC UA information models can be used to generate code. OPC UA information models are defined in an XML file and many OPC UA development tools can directly generate server code from the XML. In addition to generating server code, clients can identify the server's information model and generate the required information structures which helps developing the client application.

3.3 Nodes and Their Attributes

3.3.1 Different NodeClass Types

The address space meta model offers a strictly defined and a non-extendable group of NodeClass types. Each NodeClass type has a defined function to display defined information at runtime. NodeClass is a description of a node that defines the allowed attributes and references. Each node is a single instance of one of the NodeClass types. (Object-Oriented Internet, n.d.)

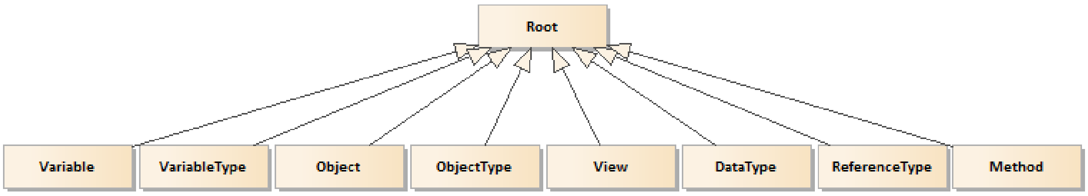

Image 4. NodeClass types (Object-Oriented Internet, n.d.)

NodeClass types:

- Variable node: A node that represents a variable, such as a sensor's measurement data.
- VariableType node: A node that offers a type definition for a variable node.
- Object node: A node that represents systems, system components or real-world objects such as a single sensor.
- ObjectType node: A node that offers a type definition for an object node.
- View node: A node that defines a node group within an address space.
- DataType node: A node that defines the data type of a variable nodes value.
- ReferenceType node: A node that defines the meaning of a reference between two nodes.
- Method node: A node that offers executable operations.

3.3.2 Attributes and Their Significance

Attributes are used to describe the nodes and the node´s attributes are determined by the NodeClass of the node. Client software's get access to a node´s attributes by using a read- or a write function. Node´s attributes are determined by it´s NodeClass but for example Node-ID and NodeClass attributes are mandatory and therefore always present. For example, a variable node typically has attributes such as value and data type. (Tulka, 2022)

3.4 Hierarchies and References

References describe the relationship between two nodes. Like attributes, references are determined as important components of a node. Unlike attributes the references are defined as node instances of ReferenceType. (Tulka, 2022)

In OPC UA references are split into two categories: hierarchical and non-hierarchical references. References are divided into two top categories, so they provide the clearest possible description of the relationship between the two nodes. Hierarchical references are used to describe that a sensor is part of a larger things, such as a robot gripper. Non-hierarchical references are used to describe that a sensors value is an attribute of a sensor node even though it’s not part of the hierarchically organized structure.

3.5 The Extensibility of Information Models

In OPC UA the information models are highly scalable and extendable. OPC Foundations website contains numerous ready to use information models and companion specifications for many different domains even including home kitchen automation. Existing information models and companion specifications can also be used as a foundation and be expanded upon to create your own information model. OPC UA information modelling has an object-oriented structure and therefore expanding existing information models is easy since you can inherit an existing type from the information model and add for example desired attributes. Information model´s hierarchic structure makes it so extending it is possible without breaking the existing base structure.

4 AUTOMATION SYSTEM INTERFACES

4.1 Benefits of OPC UA in Interface Design and Integration

OPC UA information models offer a standardized structure to reveal information which reduces the amount of inconsistency in interfaces. In interface design, standardization reduces integration workload and errors. OPC UA information models don’t rely on raw data alone but also include semantic context such as the unit of measure. The semantic context helps interface designers create user friendly interfaces where the data is directly utilizable without the need for manual interpretation. OPC UA´s expandable information models enable modular development of interfaces. Modular development means that new functionality can be added without breaking the existing interfaces.

5 BENEFITS AND CHALLENGES

5.1 Advantages of Information Modelling for Automation Systems

The benefits and advantages of OPC UA information modelling can be divided into three categories: standardizing, contextualization and futureproofing. (HMS, 2024). Standardization is useful because the information model uses a consistent group of variable types, objects and references to describe devices. This makes it easy to integrate new and different applications and systems since they can all use the same information model to understand and exchange information. (HMS, 2024). Standardization also makes it possible to replace any modelled device within the system since the interface stays the same. (Prosys OPC Ltd, 2023).

OPC UA information models offer contextualized information because the purpose and context of the data is modelled as well instead of just the structure and format of the data. The data´s purpose and context are well clarified by the object-oriented like modelling. The benefits of contextualization come apparent when the contextualized data is analysed. Contextualized data can be used to for example optimize processes, preventative maintenance and quality control. (HMS, 2024)

The third benefit of information modelling is that its future proof. Information models are future proof because OPC Foundation has offered companies a possibility to create customized information models. Another contributor to the future proofing is the possibility to extend information models. Extending information models allows the addition of features without interfering with the underlying systems. (HMS, 2024)

5.2 Challenges in OPC UA Information Modelling

Even though OPC UA´s information modelling offers a lot and bring numerous benefits it also has a clear challenge – the learning curve. OPC UA standard´s complexity and extent can cause challenges for new users. Effective use of the standard requires thorough familiarization and research.

In addition to the learning curve one of the key challenges is translating the modelling instructions into formal, programmatically readable rules. Now, most of the standards are only available in the form of an unstructured PDF-document. This creates the assumption that the person designing the information model is deeply familiar with the standard, which can be hundreds of pages long, and can successfully apply the guidelines it sets out. In practice this assumption is unrealistic and often leads into expensive issues with compatibility. Because of this, automation would be needed that could easily and reliably check that the information model complies with essential standards. (Bareedu etc., 2024, s. 1-3)

6 FUTURE DIRECTIONS

6.1 Automation of Information Modelling

OPC UA information modelling offers the possibility of displaying the semantic structure between the industrial devices, which makes the data compatible with different systems (Bareedu etc., 2024, s. 4). OPC UA is structured in layers, separating its core functionality from technological implementations. This design enables the standard to be adapted to future network technologies. (OPC Foundation, 2024a)

With automatic information model validation, it can be ensured that these models follow the OPC UA standards. Validation reduces errors and makes the integration process easier.

Semantic validation focuses on the model’s compatibility with the standards. For example, according to the Machinery-standard YearOfConstruction-variable´s value must be a 4-digit number but this restriction is only documented in writing and it cannot be verified automatically without formal rules (Bareedu etc., 2024, s. 4-5). To avoid inconsistencies like this the OPC UA work group for semantic validation has developed methods to transform restrictions into machine interpretable rules, such as SPARQL queries (Bareedu ym., 2024, s. 10).

6.2 OPC UA FX (Field eXchange)

OPC UA Field eXchange also known as OPC UA FX is an extension of OPC UA communication standard. It is developed to suit the needs of factory´s communication requirements. OPC UA FX takes OPC UA standard a step forward by bringing low level extensions that respond to the requirements for communication within the machines and the synchronization between machines regardless of their manufacturer. The combination of OPC UA and OPC UA FX aims to reach unified data transfer and display through all layers; all the way from a single sensor to the cloud. (HMS, 2023)

7 IMPLEMENTATION

7.1 Overview of the Custom Information Model

Implementation of a custom information model starts by defining what the model is for. The clearer the model´s use case and requirements are the easier the development of the custom information model is. The information model should clearly solve a problem and provide useful context for the system.

The information model created during this thesis is for Cimcorp MBR gantry robot family systems. The Cimcorp MBR gantry robot family is a robot system engineered for flexible and high-speed automated material handling in tire manufacturing and distribution logistics (Image 5). MBR gantry robots have spans up to 100 meters long, 14 meters wide and payload capacities tailored from a single plastic crate to heavy-duty truck tires. Cimcorp uses Bosch Rexroth´s ctrlX AUTOMATION platform within the MBR gantry robot systems. Bosch Rexroth ctrlX CORE PLC´s host an OPC UA server that supports the possibility for custom information models.

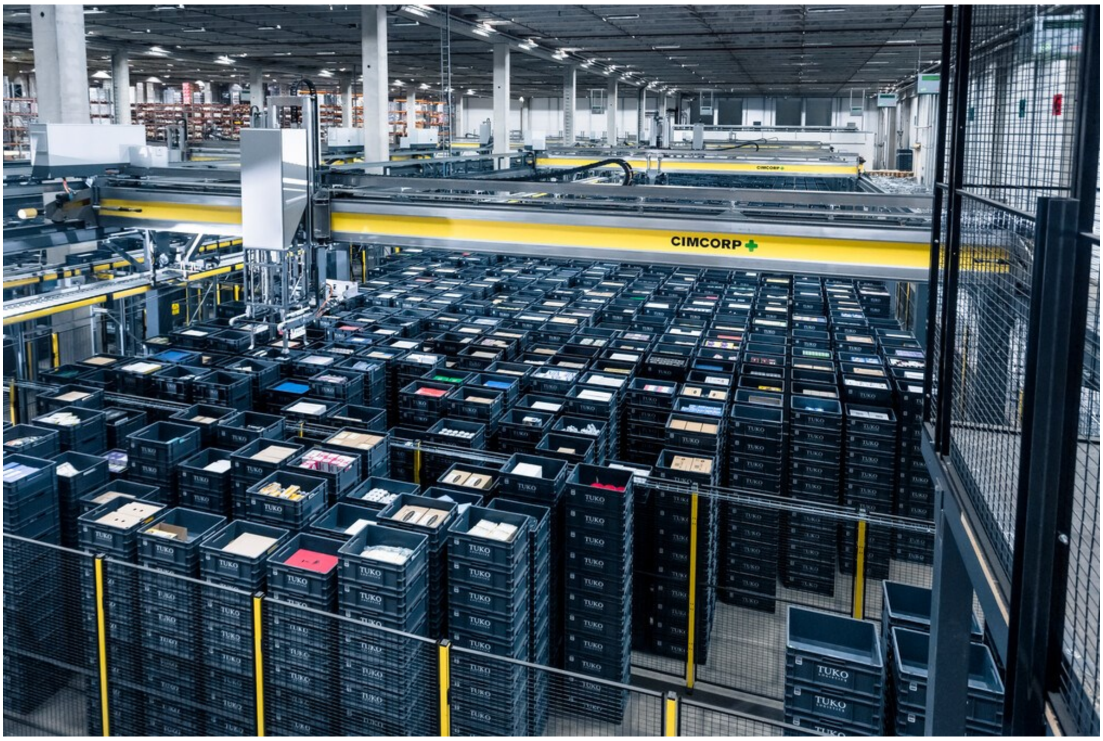

Image 5. Gantry robot cell (Cimcorp, n.d.)

As stated previously, custom OPC UA information models should solve a problem. The information model created during this thesis aims to solve multiple problems at once. One of the problems is that the PLC creates a rather unstructured interface which adds unnecessary complexity for the API development. The information model also makes development of new features easier since the features can be added on top of a structured interface in a way that does not interfere with older interface versions. The information model is also expected to reduce the engineering overhead during new projects. The information model reduces the engineering overhead by providing a standardized interface for all the MBR product variations.

The custom interface solves multiple problems, so it also supports multiple use cases. The most apparent use case is describing the interface in an understandable manner by providing structure and context to the data. Other use cases are more tailored towards versioning and other engineering related areas.

7.1.1 Design Process

The design process of the custom information model started with requirement analysis. Initial requirement analysis was discussed with my supervisor, and we decided to start with broad requirements since OPC UA information modelling was a rather new concept. It was identified that providing a user-friendly interface description was the main requirement of the information model and other more specific requirements will be defined as needed. During the design process, many of the design principles came straight from OPC UA information modelling best practices published by OPC Foundation.

7.1.2 Model Structure

The information model was implemented in its own namespace to ensure uniqueness of the namespace index. The namespace was divided into a few top-level categories in a modular fashion. These top-level categories are already a part of the type hierarchy.

The information models type hierarchy is strong and established all the way forward from the model's root node. This way the information model has a consistent structure regardless of the specific device features or variations. The deeply nested type hierarchy utilizes OPC UA addons and modelling rules to maintain user-friendly and intuitive hierarchy.

The information model gets its structure from the reference types used to connect nodes. The information model utilizes many of the reference types provided by the OPC Foundation standard specification. Parts of the system are described using the HasComponent reference and HasAlias references are used to provide alias names for the nodes. The custom information model implementation didn't require defining new custom reference types since the OPC UA standard already contains the typical reference types.

7.1.3 Key Components of the Model

The information model is built based on the system architecture so that reflects the key functional and structural entities of the automation system. The design of the model emphasizes a logical and hierarchical structure for optimal readability by humans. The information model structure also supports further expansion and adding additional semantic data to the model in accordance with the OPC UA standard.

The information model is based on describing properties and states as abstract data types, enabling consistent representation across device levels, software components and control logic. This allows the formation of a shared framework between system interfaces. A shared framework for all system interfaces promotes data exchange and interoperability

Structurally the information model utilizes a deeply nested type hierarchy that contains most of the entire system. This type hierarchy consists of type definitions, bit field structures, enumerations and union structures. This way the information mode describes the states of devices, functions and processes while providing a clear to interpret and logical address space for humans.

The information model forms a unified and extensible data structure that supports communication and semantic consistency between different parts of the automation system, without being tied to specific control modules or device implementations.

7.2 Implementation Details

Before the implementation phase the development tools were selected. The requirements for the modelling tool consisted of three criteria. First requirement was that the model needs to be edited somehow and an OPC UA information modelling software featuring a GUI was selected for this task. The second requirement was automatic code generation based on the information model. Many tools exist for generating ready to use backend code from OPC UA information models, but our OPC UA Java SDK already supports code generation from OPC UA information models, so this feature wasn’t required from the modelling software. The third requirement for the development tool was features for validating and testing the model.

Creating a custom OPC UA information model can be accomplished with nothing but a basic text or code editor, however that becomes really complicated fast when working with larger models. As stated previously it was decided that a modelling software featuring a GUI will be used. There are a few options when it comes to selecting an OPC UA information modelling software within our requirements.

First the modelling softwares were compared based on the available features and price. UaModeler was selected as the modelling software and the license was acquired from our partner company. UaModeler was selected because it has comprehensive model validation features and complete platform independence.

After selecting the tools used for creating the information model it was time to define the requirements. The main requirement was to create an information model that provides a clear description of the interface to a human. After acquiring the license for UaModeler, I started familiarizing myself with existing information models and companion specifications to see if a good fit already exists. Since the information model´s purpose is slightly different there wasn´t any models that could be used directly and it was decided not to import small parts of existing models as this would create a dependency for my information model.

The first step of the modelling process was to get familiar with the existing interface and all its components. After studying the existing interface the key structural properties were defined and the actual implementation started. Implementation was done using an iterative approach, where I validated the information model after making changes. Iterative development was decided to make sure the information model is always according to the OPC UA standard and passed all the validation tests. When I reached the point that I had replicated the existing interface in the OPC UA information model we had a meeting with the device development team. In the meeting the structure and contents of the information model were reviewed. The information model is currently awaiting first real-world implementation in a project scheduled for 2026.

7.3 Deployment and Use Case Simulation

The information model was created following the OPC UA information modelling best practices so integrating the model into an OPC UA server is straightforward. The information model is used in ctrlX PLC´s that support OPC UA information models so integrating the model in the server happens by creating bin files from the information model using a CLI script created by Bosch Rexroth for integrating information models into their PLC´s. After uploading the information model files to the PLC it is also necessary to map the information model nodes to the actual underlying datalayer. This is done with a JSON file where source and target nodes are defined. It is also possible to define strict typing for the mapping. Strict typing makes it so the information model and underlying datalayer nodes are not connected in case the data types are different. This is a useful feature for integration since the ctrlX OS notifies you if strict typing is not achieved, for example if the information model has a node with datatype Int32 and the same node in the datalayer is of type double.

7.4 Challenges and Lessons Learned

Custom OPC UA information models are complex and since I was not familiar with it when starting this thesis work there were many challenges and lessons learned. One of the most major challenges was the lack of examples regarding the information model´s design. Finding information to help solve complex problems was difficult and most of the time I had to rely on OPC Foundation´s documentation. Another difficulty I encountered had to do with the datatypes of the PLC variables. Many of the PLC variables were datatypes that are not natively supported by OPC UA. The most problematic datatypes were enums, unions and custom structures which I had to manually recreate in OPC UA and accept the fact that strict typing would not be achievable with these variables.

8 CONCLUSIONS AND RESULTS

The objective was to create a custom OPC UA model to describe the interface that the PLC provides. This objective was comprehensively reached. The information model was restricted to Cimcorp´s gantry robot family. This custom information model describes the interface from the PLC towards our user interface. The model´s scope was restricted to only consider the interface that already exists, and the model was implemented to add semantic context to the different variables that the PLC exposes. Implementing the model in the typical manner to describe the real underlying system and its relationships was left outside of the scope of the thesis. Approach was to create a dynamic typing system that enables to define different types of gantry family robots using the same frame.

OPC UA is widely used across the automation industry and is often the preferred choice for data exchange in automation systems. This is also the case for Cimcorp´s newest generation controllers. This thesis is a concrete step in moving into the most modern technologies. Using the newest and most popular technologies is important for any automation company not just because of the technologies themselves but also because most of the partners are also using these technologies.

This thesis also approaches OPC UA information modelling in a unique way. Typically, OPC UA information models are used to describe the physical system and its components in a hierarchical manner. The information model created during this thesis focuses on turning the interface into a hierarchical and logical to browse interface for both servers and human clients.

The information model created during this thesis was created to solve a single problem; there is no way to properly describe the interface to humans for example during the development of new features. With that requirement in mind, it was decided early on that the typical interface description with the real-world system components wasn’t suitable. Modelling real world system components was left outside of the scope of this thesis. Another limitation with this thesis was the lack of real-world implementation of the information model. This is due to tight project schedules and at the time there wasn´t really a suitable project for testing the information model. Testing of the information model in an actual project is scheduled for 2026.

The development of the OPC UA information model demonstrated how structured, standardized data representation can simplify automation systems interface design, improve long-term maintainability and interoperability. Although the implementation focused purely on Cimcorp´s MBR gantry robot family, the same modelling principles can be applied to other automation systems as well. The results show that a limited-scope model can significantly clarify the automation system´s interface. Future work should concentrate on deploying the information model in production environments and evaluating the impact on engineering efficiency and interoperability. Another track for future work is to implement the information model in the server-side code to reduce project specific change requirements. In a broad context, this work confirms that OPC UA information modelling is a practical and useful step toward more unified and future-proof automation systems in the future.

REFERENCES

Industry40tv. (2.9.2020). What is OPC UA and How it Works? [1 of 11] [video]. YouTube. https://www.youtube.com/watch?v=vRk42W_4R0o

OPC Foundation. (n.d.). Unified Architecture – Landingpage. Retrieved January 6, 2025, from https://opcfoundation.org/about/opc-technologies/opc-ua/

Real Time Automation. (n.d.). What Is Classic OPC? Retrieved December 8, 2024, from https://www.rtautomation.com/rtas-blog/what-is-classic-opc/

Tulka, T. (30.10.2022). OPC UA Basic Concepts. https://blog.ttulka.com/opc-ua-basic-concepts/

Profanter, S. (28.8.2023). OPC UA Address Space Explained. https://profanter.medium.com/opc-ua-address-space-explained-a6d7ee9f6a12

Object-Oriented Internet. (n.d.). Address Space and Address Space Model. Retrieved February 15, 2025, from https://commsvr.gitbook.io/ooi/semantic-data-processing/addressspaceaddressspacemodel

HMS. (14.1.2024). How OPC UA´s Information Models Improve Efficiency and Enable Better Decision-Making. https://www.hms-networks.com/news/news-details/14-01-2024-how-opc-ua%27s-information-models-improve-efficiency-and-enable-better-decision-making

Prosys OPC Ltd. (14.6.2023). Learn OPC UA Basics #5: OPC UA Information Models [video]. YouTube. https://www.youtube.com/watch?v=n6QBzC7cyUc

HMS. (12.10.2023). OPC UA FX – Benefits and anticipated impact on the industry. https://www.hms-networks.com/news/news-details/11-10-2023-opc-ua-fx---benefits-and-anticipated-impact-on-the-industry

OPC Foundation. (29.11.2024a). OPC 10000-1: UA Part 1: Overview and Concepts. https://reference.opcfoundation.org/Core/Part1/v105/docs/

Bareedu, Y. S., Frühwirth, T., Niedermeier, C., Sabou, M., Steindl, G., Thuluva, A. S., Tsaneva, S., & Tufek Ozkaya, N. (2024). Deriving sematic validation rules from industrial standards: An OPC UA study. Sematic Web, 15(2), 1-20. https://doi.org/10.3233/SW-233342

OPC Foundation. (29.11.2024b). OPC 10000-5: UA Part 5: Information Model. https://reference.opcfoundation.org/Core/Part5/v105/docs/

Prosys OPC Ltd. (14.12.2021). OPC UA PubSub Explained. OPC UA PubSub Explained - Prosys OPC