Journal of Industrial Information Integration 34 (2023) 100482

Contents lists available at ScienceDirect

Journal of Industrial Information Integration

journal homepage: www.elsevier.com/locate/jii

CNC Machines Integration in Smart Factories using OPC UA[☆]

André Martins[a,b,d,*], João Lucas[a,e], Hugo Costelha[a,b,c], Carlos Neves[a,b]

[a] *School of Technology and Management, Polytechnic of Leiria, Leiria, Portugal*
[b] *Institute for Systems Engineering and Computers - Coimbra (INESCC), Leiria, Portugal*
[c] *Institute for Systems Engineering and Computers - Technology and Science (INESCTEC), Porto, Portugal*
[d] *Technological University of the Shannon: Midlands Midwest, Limerick, Ireland*
[e] *GLN Plast, Maceira, Portugal*

ARTICLE INFO

Keywords:
Equipment integration
Smart factory
Industry 4.0
OPC UA
Machine monitoring
Computer numeric control

ABSTRACT

This paper examines the idea of Industry 4.0 from the perspective of the molds industry, a vital industry in today's industrial panorama. Several technologies, particularly in the area of machining equipment, have been introduced as a result of the industry's constant modernization. This technological diversity makes automatic interconnection with production management software extremely difficult, as each brand and model requires different, mostly proprietary, interfaces and communication protocols. In the methodology presented in this paper, a development of monitoring solutions for machining devices is defined supporting the leading equipment and operations used by molds industry companies. OPC UA is employed for high-level communication between the various systems for a standardized approach. The approach combines various machine interfaces on a single system to cover a significant subset of machining equipment currently used by the molds industry, as a key result of this paper and given the variety of monitoring systems and communication protocols. This type of all-in-one approach will provide production managers with the information they need to monitor and improve the complete manufacturing process.

1. Introduction

All developments considered within the scope of the 4th industrial revolution depend on the shop floor digitalization. In fact, the knowledge about the process, preferably obtained automatically, is the foundation of a "Smart Factory" architecture [1].

Currently, several industrial sectors, mostly formed by Small and Medium Enterprises (SME), use a direct data collection process in their manufacturing lines. However, these data collection processes are not uniform, with many companies still not investing in them [2]. Industries such as electronics and automotive, which have highly automated assembly lines, invest more in these data collection processes, resulting in more readily available process data [2].

This availability is due to the presence of specific instrumentation required for the actual process' automation. Furthermore, the automation processes may generate more useful data, including higher level of abstraction information.

The molds industry has embraced automation at an early stage, particularly using Computer Numeric Control (CNC) technologies like milling, turning, and electric discharge machining (EDM), with semi-automated connections to the machines for program transmission from machining software (DNC). In the opposite direction, i.e., collecting information regarding the process or equipment state, the current automation level is still low, and it is neither universal nor done with the same depth in most companies [3].

In the authors' opinion, this gap is primarily due to the wide range of CNC controllers on the market and the current lack of a *de facto* standard for machining interfaces. The implications of this missing

☆ This work was partially developed under the project TOOLING4G (POCI-01-0247-FEDER-024516) and the project S4PLAST - Sustainable Plastics Advanced Solutions (POCI-01-0247-FEDER-046089), supported by Programa Operacional Competitividade e Internacionalização (POCI), Programa Operacional Regional de Lisboa, Portugal 2020 and Fundo Europeu de Desenvolvimento Regional (FEDER). This project was also partially financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within projects UIDB/00308/2020 and LA/P/0063/2020, and under the Scientific Employment Stimulus - Institutional Call CEECINST/00051/2018. Special thanks to the Technological University of the Shannon: Midlands Midwest, a RUN-EU partner who also supported this work.

* Corresponding author at: School of Technology and Management, Polytechnic of Leiria, Leiria, Portugal.
E-mail addresses: andre.martins@ipleiria.pt (A. Martins), 2140879@my.ipleiria.pt (J. Lucas), hugo.costelha@ipleiria.pt (H. Costelha), carlos.neves@ipleiria.pt (C. Neves).

https://doi.org/10.1016/j.jii.2023.100482
Received 30 April 2021; Received in revised form 9 May 2023; Accepted 1 June 2023
Available online 7 June 2023

standard is exacerbated by the fact that most machining equipment is either closed to third-party applications, or only offers a vendor-specific proprietary interface, making data collection from machining processes more difficult. The main goal of this work is to create a standard monitoring solution approach for CNC-based machining processes, lowering the adoption barrier for this sector within the scope of Industry 4.0.

To accomplish the aforementioned goal, a methodology for developing monitoring solutions for machining equipment has been defined and tested. This work was based on the leading equipment and operations used in molds industry companies, using as a reference a mold making cluster in Marinha Grande, Portugal recognized worldwide[1], and based on the Open Platform Communications Unified Architecture (OPC UA) for high-level communication between the various systems. The approach explores the characteristics of the equipment and prioritizes equipment interaction through a communication protocol, Software Development Kit (SDK), or Application Program Interface (API), if one is available. When this method is not applicable due to the lack of appropriate equipment interfaces, a set of messages is defined and implemented within the CNC program code that can be exchanged with the equipment. In the worst-case scenario, if the equipment does not support any communication, the approach relies on direct access to sensors, either those are already integrated or those externally added to the CNC machine to this effect, allowing performance metrics to be collected. These various approaches are combined when needed, to obtain a more relevant data set, allowing for improved system monitoring, or even making relevant predictions regarding the machining process.

There is a significant body of knowledge related to CNC machine monitoring techniques, particularly with tool condition monitoring. The work carried by Downey et al. [4,5] collects and analyses data from three sensor technologies (force, acoustic and vibration) to monitor CNC tool wear in a real-time production environment. Other works, such as those by René de Jesús et al. [6], and Stavropoulos et al. [7], employ driver current signal analysis as a sensorless approach on tool wear and breakage detection.

Others, like Siddhartha et al. [8], proposed a system for remote condition monitoring and data-based decision-making process for CNC machines, using a Raspberry Pi to publish the machine data via the MQTT protocol. Using the publish–subscribe communication approach provides for a simple and efficient data exchange mechanism, although the authors identified issues regarding data exchange security which, in real industrial scenarios, can pose serious problems.

There are fewer implementations and results available regarding the use of the OPC UA standard to monitor CNC machines over a wider range of parameters, with the three most relevant regarding this work being described next. Mourtziz et al. [9] proposed an OPC UA-based framework for milling and lathe CNC machine tool modeling. They also created a data acquisition device, to integrate legacy machine tools with no connectivity capabilities into their holistic framework, and presented a laboratory case study to validate the proposed system. This research has made significant developments in data acquisition using external sensors, not taking into account the possibility of direct data exchange with the CNC controllers.

Martinov et al. [10] discussed the possibilities of using OPC UA to collect information from CNC machines, and graphically represent this data. However, this work only describes one implementation example, for a specific CNC machine controller, lacking tests in various scenarios, including other CNC controller models.

To enable standardized, interoperable, and efficient data communication between machine tools and various types of software applications, Liu et al. [11] propose a Cyber–Physical Machine Tools (CPMT) platform based on OPC UA and MTConnect. An OPC UA client, an Augmented Reality assisted wearable Human–Machine Interface, and a conceptual framework for a CPMT-powered cloud manufacturing environment were developed, to demonstrate the benefits of the proposed CPMT platform.

[1] https://www.moldmakingtechnology.com/blog/post/state-of-mold-manufacturing-in-portugal

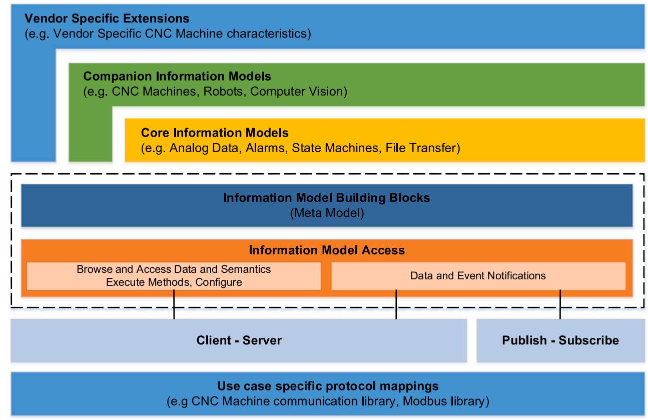

Fig. 1. OPC UA architecture.
Source: Adapted from https://opcfoundation.org/about/opc-technologies/opc-ua/.

In this paper, a new information model will be provided, extending the CNC Companion Specifications (CS) [12] with new data types relevant to industry, as a result of the collaboration with the molds industry in the regions of Leiria and Marinha Grande, in Portugal. Based on this information model, a complete OPC UA server has been developed and will be described below. It is suitable to be used both with legacy and new CNC machines, following a modular design in order to enable configuration-based addition and removal of components, thus providing for a standard approach to monitor relevant CNC-based process parameters.

The remainder of this paper starts with Section 2, providing a brief overview of the main technology used in the developed work. The definition of a methodology and the development of a CNC monitoring solution, based on an OPC UA server, are described in Section 3. Section 4 describes the case studies that were used to test the developed methodology, considering three different CNC controller vendors. The obtained results are presented in Section 5, while Section 6 presents conclusions and future work directions.

2. OPC UA-based modeling and integration

The OPC UA standard is the primary supporting technology for the developed work. This section contains a brief description of the standard, as well as its main characteristics.

OPC UA is the approach for a communication layer implementation recommended by the German Reference Architecture Model Industrie 4.0[2] (RAMI4.0) industrial platform [13], being defined by the international standard IEC 62541. Due to its characteristics, OPC UA is a standard that ensures the open connectivity, interoperability, security, scalability and compliance, of industrial devices and systems. OPC UA is much more than a communication protocol because it is based on a multi-layered architecture (see Fig. 1), in which a single server provides all information and services from a given system with active security. It includes built-in information models and defines the fundamental rules for data exchange and interfaces.

[2] https://www.plattform-i40.de/

2.1. Open connectivity

OPC UA is the successor of OPC Classic[3] and, although OPC Classic can be set up to provide a reasonable level of security, since it relies on Microsoft Windows and DCOM/COM functionalities, it can be challenging to set up in other operating systems. Conversely, OPC UA was designed to be cross-platform, working in a variety of environments and platforms, with security built-in from the onset [14].

2.2. Interoperability

To allow interoperability, OPC UA employs the Address Space concept to represent all the information generated by a specific system or device. This information is usable and can be decoded by other systems that use the same protocol. Its structure is based on information models, which results in a standardized representation of the data and information available to client systems. The object concept is used by OPC UA to represent system data and behavior. Objects are used as locations to store variables, events and methods, and they are linked together by references. The standard information model is designed to allow type definition, so that designers can meet their own application needs. Furthermore, information models for specific areas, such as CNC systems, can be combined. These specific models are called Companion Specifications (CS) and derive from the standard model, inheriting its features, but can also include modifications, which can also be combined, depending on the system needs. The CS are typically developed based on insights given by several companies and organizations in a given field, resulting in a significant contribution towards the standardization and adoption of these base information models.

2.3. Security

OPC UA places a high value on secure communication between client and server, addressing a wide range of applications, and encompassing different security and timing requirements. Table 1 shows the various security modes that can be used to enable both digital signature and encryption mechanisms, only digital signature mechanisms, or none of them. An OPC UA client can select the desired security mechanisms from a list of options provided by a given OPC UA server. After a secure channel and session have been established between server and client, there is still the need for user credentials, represented by another certificate, or by a username/password combination [15]. Given that the majority of OPC UA applications are used within industrial environments, security concerns must be addressed. The German Federal Office for Information Security commissioned the OPC UA security analysis report [16], which presents recommended measures and procedures to follow when implementing OPC UA-based solutions.

2.4. Scalability and compliance

Various OPC UA profiles are defined, as shown in Table 2, considering different application scenarios, which allows OPC UA to scale down to a chip-level using the Nano Embedded Device profile, while still retaining its main features [18]. Alternatively, in more complex scenarios, a PC-based server using the Standard UA profile provides for additional functionality. The OPC Foundation developed the UA Compliance Test Tool (UACTT),[4] as a way to determine which profiles each application supports, and to ensure that the designed OPC UA server or client complies with the OPC UA specification.

[3] https://opcfoundation.org/about/opc-technologies/opc-classic/

[4] https://opcfoundation.org/developer-tools/certification-test-tools/opc-ua-compliance-test-tool-uactt/

Table 1
OPC UA security modes [17].

Mode	Properties
None	No security.
Sign	Encoded with sender's private key. Only certificate owner has the private key. Anyone can verify the identity. Provides authenticity.
SignAndEncrypt	Messages are signed and also encrypted. Encoding with receiver's public key. Anyone can encrypt. Only the certificate owner can read. Authenticity, confidentiality and integrity.

2.5. Communication mechanisms

Regarding information exchange, OPC UA provides two main communication mechanisms: Client–Server and Publish–Subscribe (PubSub) [20]. In the Client–Server mechanism, the client accesses the information provided by the server through defined services. The PubSub mechanism can be applied in different ways: for messaging over local area networks, data is published by the OPC UA server (publisher), and consumed by multiple authorized OPC UA clients (subscribers), using Time Sensitive Networking to allow real-time low delay communications; for messaging over global networks (WAN/Cloud), the OPC UA PubSub specification [21] defines mappings on existing protocols such as, for instance, MQTT.[5]

2.6. Information integration

There are several uses that can be given to raw data gathered from CNC machines. Some of the works available describe scenarios where such data is used for fault detection and predictive maintenance [22–25]. In such scenarios, the main goal is to convert CNC raw data into useful insights for the operator and production manager. This, in turn, leaves us with the perception that most of these works focus on the analysis using Artificial Intelligence (AI) techniques and Machine Learning (ML) methods, giving less relevance to the way this data is made available from the machines. Although, when we discuss the broader topic of information integration from industrial devices, such as CNC machines, we are not only talking about conclusions from AI and ML. Also, given the effort needed to gather data from some legacy industrial devices, it is important to have other future uses in mind when performing this task, preparing them for integration across several Smart Factory environments. As an example of integration challenges in such environments, we can refer to the interaction requirements for applying the Plug-and-Produce concept. This concept, derived from the well-known Plug & Play approach in computer systems, aims at the factory's flexibility to adapt to new production requirements due to rapidly changing market demands. Therefore, production systems and shop floors must be prepared for higher flexibility and reconfigurability to adapt to these circumstances. To achieve automatic configuration and information exchange without the need for reprogramming automation tasks, one of the basic requirements is a generic standardized component interface [26]. The following Section of the presented work describes the development of an information integration procedure for CNC machines, using OPC UA as the foundational technology, thus providing machines with a standardized interface for further uses in Smart Factory environments.

[5] http://mqtt.org/

Table 2
OPC UA profiles [19].

Profile	Characteristics
Nano Embedded Device	Limited functionality, for very small devices, e.g. sensors. Only one connection, without message signing or encryption, subscriptions and method calls.
Micro Embedded Device	Restricted functionality, with at least two parallel connections, additional subscriptions and data monitoring, but no UA security or method calls.
Embedded UA	Basic OPC UA functionalities are available, plus UA security and method calls.
Standard UA	Includes all functionalities for secure information access, including UA security. No alarms and no history. PC-based servers should support at least this profile.

3. Developing an OPC UA server for CNC machines

Direct interaction with different CNC controllers has become extremely difficult due to the current market diversity, and given that some of this equipment is not open to third-party applications. However, recently released equipment frequently has a communication protocol available, with some vendors providing an SDK, an API, or simply the definition of a set of messages that can be exchanged with the equipment for this purpose. Alternatively to the communication-based data acquisition from the CNC controllers, direct access to the sensors, already integrated, or externally added to the CNC device, allows collecting performance metrics, which in turn provides the necessary data for predicting the machine and/or process states over time.

This section describes a methodology for developing a CNC monitoring solution based on an OPC UA server, along with the development of a generic C# class, while providing the details concerning the data exchange with the CNC machines.

3.1. Generic C# class

To allow a common generic interface to work with CNC machines from different vendors using the various approaches mentioned earlier, a generic C# abstract class (CNCBase) with abstract methods was developed, supporting the development of monitoring applications. In this approach, each CNC machine-specific brand will have its own class implementation, overriding the generic class with its own specific functions, given its own specific communication interfaces, while externally providing the information made available using the same information model, as shown in Fig. 3.

This approach simplifies the future integration of other CNC machine models. A configuration file, which allows the definition of application parameters, such as the CNC machine model/brand to be monitored, the communication address definition to be used, as well as the machine parameters and the corresponding monitoring frequency, is another important component of the developed application. The primary goal of this configuration file is to define run-time function parameters, without having to recompile the server application for a specific machine, or, for instance, whenever different data is to be acquired, or is to be acquired at different sampling rates. Entries in this (XML) configuration file follow the following syntax:

`<add key="Name" value="NodeID,Enabled,Time"/>` where

- `Name` – Method name;
- `NodeID` – Node number from the address space where the value is updated;
- `Enabled` – Parameter which enables this entry;
- `Time` – Value which represents the time interval between variable updates.

As an example, one possible entry from the XML configuration file could be `<add key="Get_x_dir_actpos" value="6008, true,1"/>`, which specifies that the `Get_x_dir_actpos` method is to be called to update every 1 second the value of node with id number `6008`.

Another relevant part of this configuration file can be found below (the modules associated with these parameters are detailed in Section 4), where it is possible to specify a set of configuration parameters for each application, such as:

- `CNC_CONTROLLER` – Allows the specification of the CNC controller model, used internally to choose the methods corresponding to each specific CNC brand and model;
- `CNC_IP` – Allows the specification of the CNC controller IP address which the developed application will connect to;
- `CNC_PORT` – Specifies the CNC controller port which the developed application will connect to ;
- `POWER_METER_IP` – Specifies the power meter IP address, which the developed application will connect to;
- `POWER_METER_PORT` – Specifies the power meter port, which the developed application will connect;
- `TURCK_IP` – Specifies the Turck module IP address, which the developed application will connect to;
- `TURCK_PORT` – Specifies the Turck module port, which the developed application will connect to;
- `VIDEO_IN_PATH` – Allows the definition of the path to a video stream source;
- `VIDEO_OUT_PATH` – Allows the definition of the path to save the generated videos;
- `VIDEO_FPS` – Specifies the generated videos frames per second;
- `VIDEO_LENGTH` – Specifies the stored video length (in seconds);
- `INFLUXDB` – Allow enabling the use of an InfluxDB local database.
- `RFID_COM_PORT` – Specifies the RFID reader COM port number, which the developed application will connect to;
- `CNC_COM_PORT` – Specifies the CNC COM port number, which the developed application will connect to.

This generic class was initially developed independently of OPC UA, allowing it to be used even if there is a paradigm shift in terms of its usage in these applications. For instance, the first version of the CNCMonitor application was designed to store the data collected from a CNC controller directly into an InfluxDB Time Series Database (TSDB).

3.2. Software-based design of the OPC UA information model

As stated previously, OPC UA was designed as a replacement of OPC Classic. One of the most important enhancements in OPC UA is a powerful Information Model concept: the Address Space. OPC UA allows real-time process data and underlying infrastructure to be exposed as a consistent information model composed of nodes. Nodes, attributes, and their mutual relationships make up the process model. This powerful concept allows one to expose not only raw process data, but also entire consistent information sets about the process state and behavior by using OPC UA. Because of its flexibility, OPC UA allows exposing even the most complex systems, however, such flexibility leads to challenges during design, development, and deployment. To better address these challenges, it is important to use software specifically designed for information model design, such as:

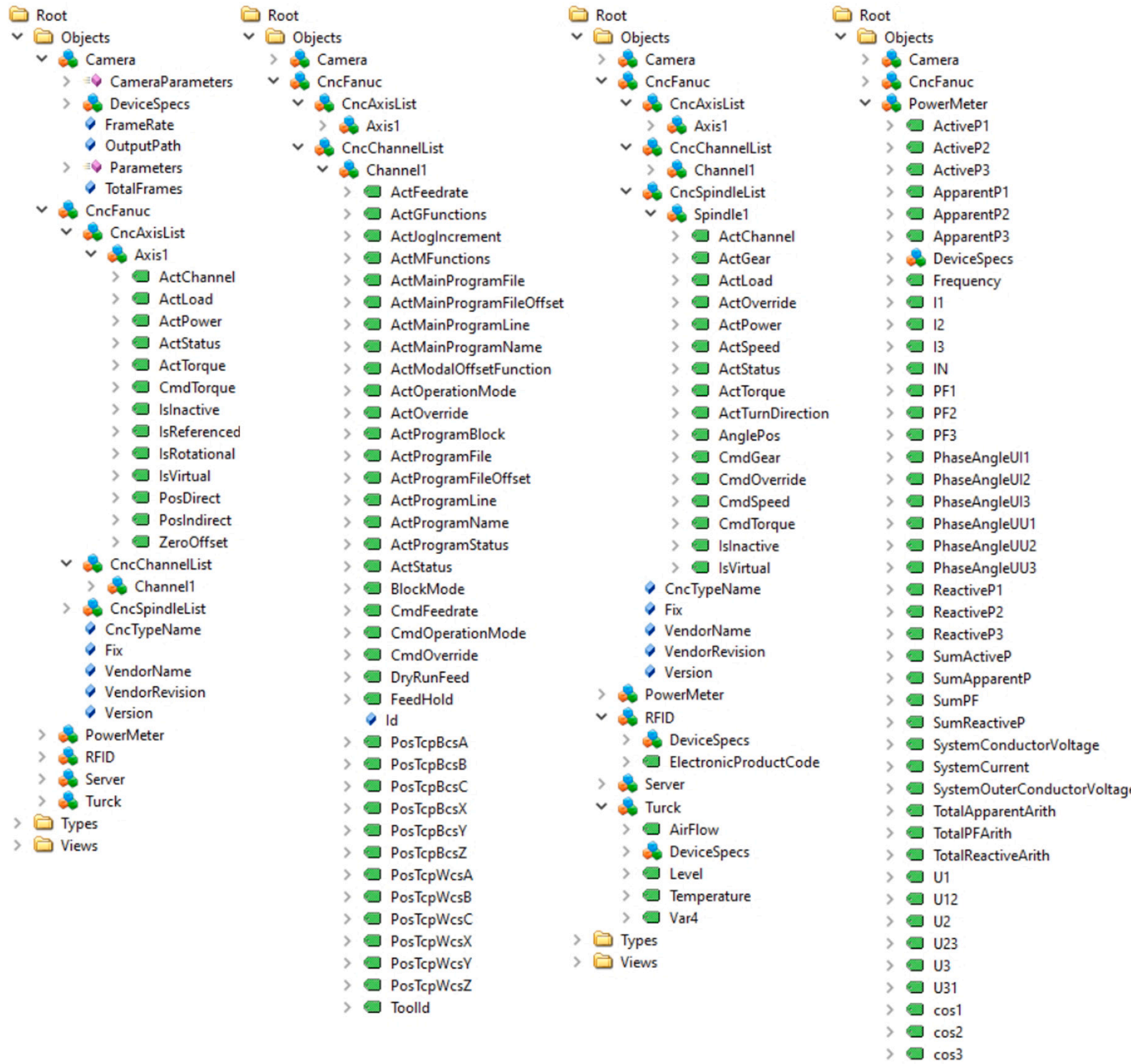

Fig. 2. Four views of the same CNC OPC UA server address space, with different nodes expanded in each view, obtained using UA expert.

- **UAModeler**[6] from Unified Automation — Requires an account to download the software, with the free version not allowing C# code generation and support for <UANodeSet> files (official format for files containing sets of OPC UA nodes);
- **UA Model eXcelerator Professional (UMX Pro)**[7] from Beeond — This tool enables developers to graphically configure an OPC UA compliant information model, produce, merge and manipulate <UANodeSet> files and generate SDK independent code;
- **OPC-UA-Modeler**[8] from Fraunhofer IOSB — Web-based graphical tool to develop information models;
- **Free OPC UA Modeler**[9] – Free and open source tool, that operates and outputs <UANodeSet> file types;
- **Siemens OPC UA Modeling Editor (SiOME)**[10] — Free tool, created by Siemens, to define OPC UA information models, and to map existing companion specifications on a SIMATIC PLC. This tool allows to import and edit information models as XML files, and both to generate and export individualized models;
- **Object Oriented-Internet Address Space Model Designer (ASMD)**[11] from CAS — Initially a commercial product of CAS Lodz Poland, this now open source information model design tool, embeds the ModelCompiler[12] module, from the OPC Foundation, and allows working with <ModelDesign> file types, and to generate <UANodeset> files, C# and ANSI C source code.

Primarily, and because the available software was either not open source, or the trial versions were very limited functionality-wise, a generic C# abstract class (CNCBase) was developed, as described in Section 3.1. It provides abstract methods to model the interface for a generic CNC machine, allowing the development of monitoring applications that work with various CNC machines. With this type of implementation, each CNC machine brand-specific class then overrides the generic one with its own functions, from its own specific communication interfaces, while still exchanging the information available using the same information model.

[6] https://www.unified-automation.com/products/development-tools/uamodeler.html
[7] https://beeond.net/umxpro/
[8] https://www.iosb.fraunhofer.de/servlet/is/35891/
[9] https://github.com/FreeOpcUa/opcua-modeler
[10] https://support.industry.siemens.com/cs/document/109755133/siemens-opc-ua-modeling-editor-(siome)-for-implementing-opc-ua-companion-specifications?dti=0&lc=en-WW
[11] https://github.com/mpostol/ASMD
[12] https://github.com/OPCFoundation/UA-ModelCompiler

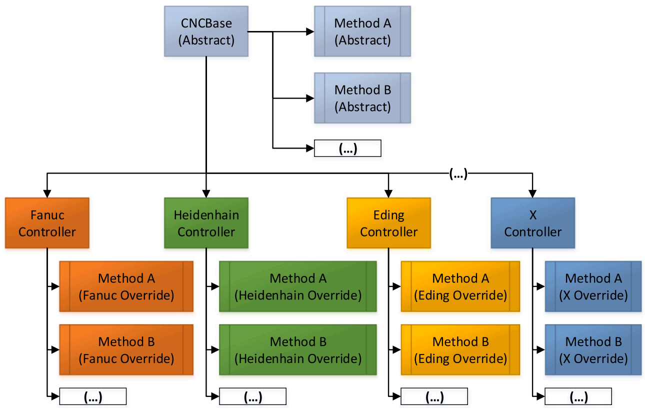

Fig. 3. Generic C# abstract class and methods.

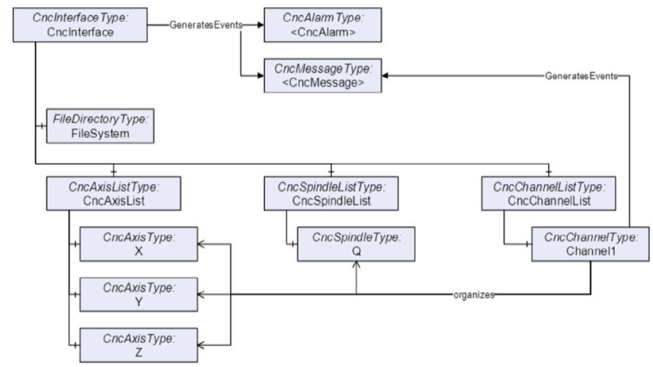

Fig. 4. 3-Axis machine tool OPC UA information model [12].

Given that the CAS' founder and Executive Director decided to open-source the ASDM software, this software was explored and used to modify the information model, and to generate the files needed to build the address space of the OPC UA servers built in this work. Hence, a new CNC machine information model was built based on the CNC CS, replacing this CNCBase functionality. Regarding the machine to sensor data connection, the C# CNCBase class is still being used.

3.3. OPC UA server

The OPC UA protocol was chosen with the goal of making the collected data available in a standardized approach. OPC Foundation offers a variety of implementations in different programming languages. Among these, the .NET stack version[13] was chosen, currently with a license that allows for developments within education and research. Among the software stacks backed by the OPC Foundation, the .NET version of OPC UA is the one with the most support and available implementations. Furthermore, the use of this Framework allows OPC UA servers to be developed on Windows, Linux and macOS (using .NET Core[14]), as well as in embedded systems applications (using the nanoFramework[15]). The developed application is based on the Reference Server,[16] which has been certified for compliance by UACTT.

The CNC CS [12] was considered when defining the Address Space of the built OPC UA Server, explicitly following the OPC UA information model example of a 3-axis machine tool, as can be seen in Fig. 4. Following a modular approach in the OPC UA server development, the number of main OPC UA objects that are available depend on the peripheral equipment connected to the computer running the server. According to the diagram from Fig. 2, currently the system supports the following objects (full details on the usage of the corresponding modules will be given in Section 4):

- `Camera` – This OPC UA object contains an OPC UA method which allows to configure the video output path, the total length of the video and the frames per second, and can be used as a trigger to generate a video. Its configuration is based on the `VIDEO_IN_PATH`, `VIDEO_OUT_PATH`, `VIDEO_FPS` and `VIDEO_LENGTH` parameter values defined in the configuration file;
- `Cnc<Model>` – The structure of this OPC UA object follow the CNC CS 3-axis machine example shown in Fig. 4. The associated data variables are updated based on the information obtained from the CNC controller. This object is added to the server and configured according with the parameter `CNC_CONTROLLER` value in the configuration file;
- `PowerMeter` – This OPC UA object contains the OPC UA variables that are mapped from the Power Meter Modbus registers. This object is added to the server if the `POWER_METER_IP` and `POWER_METER_PORT` parameters are defined in the configuration file;
- `RFID` – This OPC UA object contains a variable that is supplied with information from an external application that interacts with the RFID reader. This object is added to the server if the `RFID_COM_PORT` parameter is defined in the configuration file;
- `Turck` – This OPC UA object contains OPC UA variables for each analog input of the Turck gateway, which are mapped to the information obtained from the Turck module Modbus registers. This object is added to the server if the `TURCK_IP` and `TURCK_PORT` parameters are defined in the configuration file;

The information model to describe the CNC machines was developed using UAModeler[17] from Unified Automation. This tool provides a graphical and hierarchical representation of the designed model, following the OPC UA notation and syntax.

3.4. Data exchange with the CNC controller

To exchange data with the CNC controller, preference was given to the use of the CNC controller's communication protocol, SDK, or API, when available, for decreased deployment costs and time. If the above methods are not supported but a log system is available, a group of messages is defined and implemented within the machining code, which can be sent from the controller during the machining processes. For instance, in the FANUC case, this can be automated by adding DPRNT instructions to the ISO code within the operation of the CNC machine post-processor phase. Other vendors provide for a similar functionality.

In addition to the aforementioned data acquisition methods, or when the device does not support any communication, direct (hardware-based) access to the sensors already integrated or externally added to the CNC machine can be used to obtain the required information. In these cases, an approach was proposed which relies on devices that can support both wireless and/or cabled networks. This approach allows to benefit from both solutions [27], while keeping in mind the stringent requirements often imposed by industrial environments [28].

[13] https://github.com/OPCFoundation/UA-.NETStandard
[14] https://docs.microsoft.com/en-us/dotnet/core/
[15] https://nanoframework.net/
[16] https://github.com/OPCFoundation/UA-.NETStandard/blob/master/SampleApplications/Workshop/Reference/README.md
[17] https://www.unified-automation.com/products/development-tools/uamodeler.html

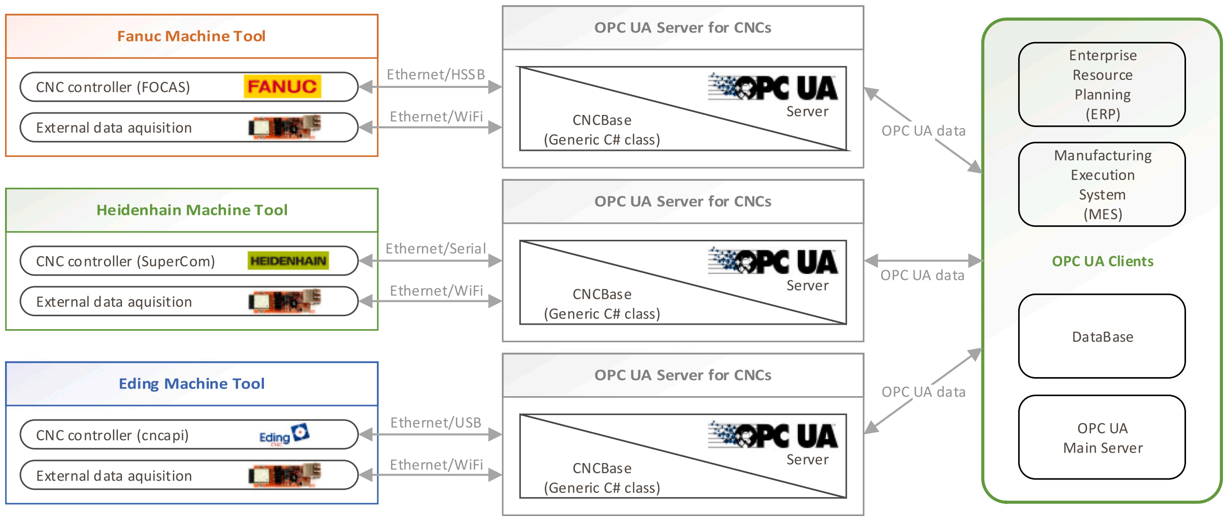

Fig. 5. Functional diagram and interaction between systems.

4. Case studies

This section provides further details about the methodology applied to specific scenarios, considering three different CNC controller models. These were chosen based on the leading equipment and operations used by molds industry companies in the Marinha Grande cluster, Portugal, and the Advanced Robotics and Smart Factories research laboratory at the Polytechnic of Leiria. Given the cluster inputs, two types of CNC controllers were chosen, namely the FANUC 31i-A and the Heidenhain iTNC530 CNC controllers, as shown in Fig. 5, as well as an Eding CNC controller, as part of a parallel project related to a CNC retrofitting process in the aforementioned laboratory.

4.1. FANUC CNC machine

Concerning data exchange with the 31i-A FANUC controller, the FANUC Open CNC API Specifications (FOCAS) library is used. This library allows data to be accessed via Ethernet or High-Speed Serial Bus (HSSB) from the CNC and Programmable Machine Control (PMC) controllers [29]. The FOCAS library, in both the 1 and 2 versions, contains numerous functions[18] that enable data exchange with FANUC CNC machines [30]. This library, however, is not supported by all the FANUC controller families, being restricted to machines from the i series. This library can be used to develop Windows and Linux-based applications, as well as applications for Android and iOS mobile platforms.

Furthermore, FANUC controllers provide a number of commands that can send variable values and various characters to external devices through the RS232 or embedded Ethernet ports. There are four of these commands, which are known as External Output Commands: POPEN (Port Open) and PCLOS (Port Closed) are used to “connect” and “disconnect” from the output port, respectively; between these two instructions, the BPRNT and DPRNT commands can be employed to output data in binary or ASCII format, respectively [31]. When the controller does not support the FOCAS library or when the FOCAS library does not provide access to the needed information, these commands can be added to the ISO CNC code in the post-processor phase, allowing the acquisition of machine parameters.

[18] https://www.inventcom.net/fanuc-focas-library/general/general

4.2. Heidenhain CNC machine

The SuperCom Heidenhain communication library[19] was used to exchange data with the Heidenhain iTNC530 controllers. Similar to FOCAS for FANUC controllers, this Adontec library allows communication with Heidenhain TNC controllers via the serial or Ethernet (TCP/IP) ports. The functions in this library are used to create data connections to one or more Heidenhain controllers. SuperCom is an event-driven application that allows transfering files to and from the Heidenhain TNC, list, create and delete folders, rename and delete files, read TNC configuration data, retrieve machine status, machine data and process data, as well as read and write memory registers, among other functionalities. Depending on the controller series, the data that is available may differ. Direct access functions are also included in the library, which can be used to retrieve or modify data directly from the controller-connected PLC memory. This library can be used to develop both Windows and Linux-based applications, although a separate license is required for each platform.

4.3. Eding CNC machine

Another project where this development was used [32] involved retrofitting a CNC machine using an Eding controller board.[20] In this case, data is exchanged with the controller using the cncapi[21] communication library over an USB or Ethernet (TCP/IP) interface. This library shares its internal functions, allowing to send and receive data, while also being used by the graphical user interface to view and control the Eding CNC controller. An OPC UA server was implemented using the methodology described above to work in parallel and communicate with the Eding software. This library allows data to be exchanged with all Eding controllers, regardless of the controlled equipment. The library, however, is only available for Windows. The Eding library is only available in C++ and is not available in C#. As a result, an existing C++-based code wrapper for C#[22] was used to allow data to be accessed from the developed C#-based OPC UA server.

[19] https://adontec.com/
[20] https://www.edingcnc.com/products.php
[21] The cncapi and its documentation is only available after having installed the Eding CNC Software.
[22] https://www.oosterhof-design.com/cncapi-netframework/

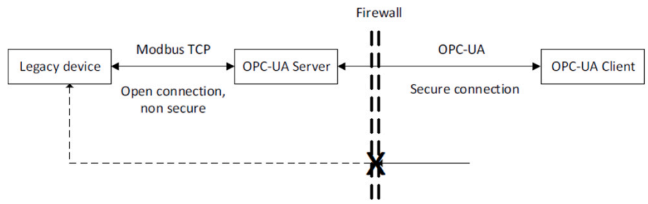

Fig. 6. Diagram representing the communication security between devices.

4.4. External sensorization - industrial equipment

Regarding the integration of industrial devices in the developed OPC UA server, two examples were added to the system: an power meter used to monitor the CNC machine's power usage and a camera to monitor and record the machining process in case of alarm events from the CNC machines.

4.4.1. Power meter

The industrial device integrated with the developed OPC UA server, in this case, was a Phoenix Contact EMpro EEM-MA370[23] power meter. The Modbus[24] TCP protocol (a version of the Modbus protocol, designed for use over a TCP/IP network) is used to make this equipment's data available.

The Modbus protocol's specification does not include any cybersecurity features (users, passwords, certificates, or others). However, the developed OPC UA server has the ability to provide these security services, required for safe communication with an external client.

As a result, either the Modbus device and the computer with the OPC UA server must be located on a private network, protected by a firewall preventing unauthorized access from outside (see Fig. 6), or a point-to-point connection must be employed.

Regarding the communication with the power meter, the developed OPC UA application is a Modbus TCP client. For external clients, the application is an OPC UA server that safely interfaces with the power meter.

There are several Modbus libraries available, such as the open-source EasyModbus,[25] the library used in this case. Given that this communication protocol is widely disseminated in industry, this implementation is particularly useful for other situations where the industrial device to be integrated supports this communication protocol.

4.4.2. Camera

The use of a camera is another example of industrial equipment integration. Currently, the camera's main goal is to acquire video footage of the machine's operation (at a lower quality and only for less detailed remote monitoring), generating videos of the process preceding each event generated by OPC UA and associated with CNC alarms. The system uses a C# linked list containing the last frames of the capture video to continuously store the last seconds (configurable) in a First-In, First-Out (FIFO) queue. When a CNC alarm event occurs, a method is invoked, and the most recent stored frames, those that contain the recording of what occurred just before and up until the event was captured, are saved to a video file and stored in a shared folder. This method also returns the path to the saved video, allowing the operation manager and client to access the alarm event and corresponding video through the OPC UA server.

A Genie Nano C1920[26] (industrial camera) from DALSA Teledyne, and an IP network camera available through an online internet stream, were tested with the developed system. This system, however, can access the video stream from any camera that supports Real-Time Streaming Protocol (RTSP) or USB Video Class (UVC), among other protocols.

Because this system also supports industrial cameras, computer vision techniques can be used to process the recordings in future developments of this work, allowing other types of information to be gathered.

4.4.3. RFID reader

Being able to integrate an RFID reader in the system, allows for the identification of the part that is being processed inside the CNC machine, as well as for operator identification. The module used is an UHF RFID Reader CF-RU5202[27] from Chafon, which is connected to the computer through USB, with Chafon providing an SDK for application development.

4.4.4. Turck gateway

For external sensorization, when industrial sensors are already installed on the CNC machine, gateways can be employed to connect these sensors with the developed system. An example of such module is the TBEN-S2-4AI[28] from Turck, which has four analog inputs, and provides access to their values through Modbus TCP/IP communication. These analog input values are accessed through Modbus registers, converted into sensor units, and then mapped to OPC UA server variables.

4.5. External sensorization - development of dedicated hardware

The Olimex ESP32-PoE IoT development board[29] was used to collect additional data. Built around the ESP32-WROOM-32 module [33], it supports WiFi, BLE, and 100 Mb Ethernet with Power-Over-Ethernet (PoE). It also has a LiPo battery connector, a MicroSD card slot, General Purpose Input Output (GPIO) headers, and a Universal Extension (UEXT) connector that allows the connection of devices that communicate via I2C, SPI, or RS232. This board is also available in a galvanic isolated version for the PoE power, as well as an industrial version[30] that supports an extended temperature range from −40 °C to 85 °C.

It is possible to create OPC UA servers for devices like this, with more limited resources, by combining this hardware with the open62541 server (an open source version of OPC UA in C). These servers provide data from a given sensor, or set of sensors, to the main server. The project from the opcua-esp32[31] repository was used as a starting point for the development of these OPC UA servers. It employs the open62541 library to build an OPC UA server for use on an ESP32, with an information model that includes a temperature sensor (DHT22), a callback method to call the ESP32 on-board LED, and two relays that can be controlled remotely as OPC UA objects.

Two implementations were developed: one which has an LM35 temperature sensor connected to an analog input, and another which has an MPU6050 module connected through the I2C interface, measuring data from the 3-axis accelerometer and 3-axis gyroscope. In both cases, the sensor data was made available through OPC UA to the main server.

[23] https://www.phoenixcontact.com/online/portal/us?uri=pxc-oc-itemdetail:pid=2907983

[24] https://www.ni.com/pt-pt/innovations/white-papers/14/the-modbus-protocol-in-depth.html

[25] https://sourceforge.net/projects/easymodbustcp/files/latest/download

[26] https://www.teledynedalsa.com/en/products/imaging/cameras/genie-nano-1gige/

[27] https://www.chafon.com/productdetails.aspx?pid=535

[28] https://www.turck.us/en/productgroup/Fieldbus%20Technology/

[29] https://www.olimex.com/Products/IoT/ESP32/ESP32-POE/open-source-hardware

[30] https://www.olimex.com/Products/IoT/ESP32/ESP32-POE-ISO/open-source-hardware

[31] https://github.com/cmbahadir/opcua-esp32

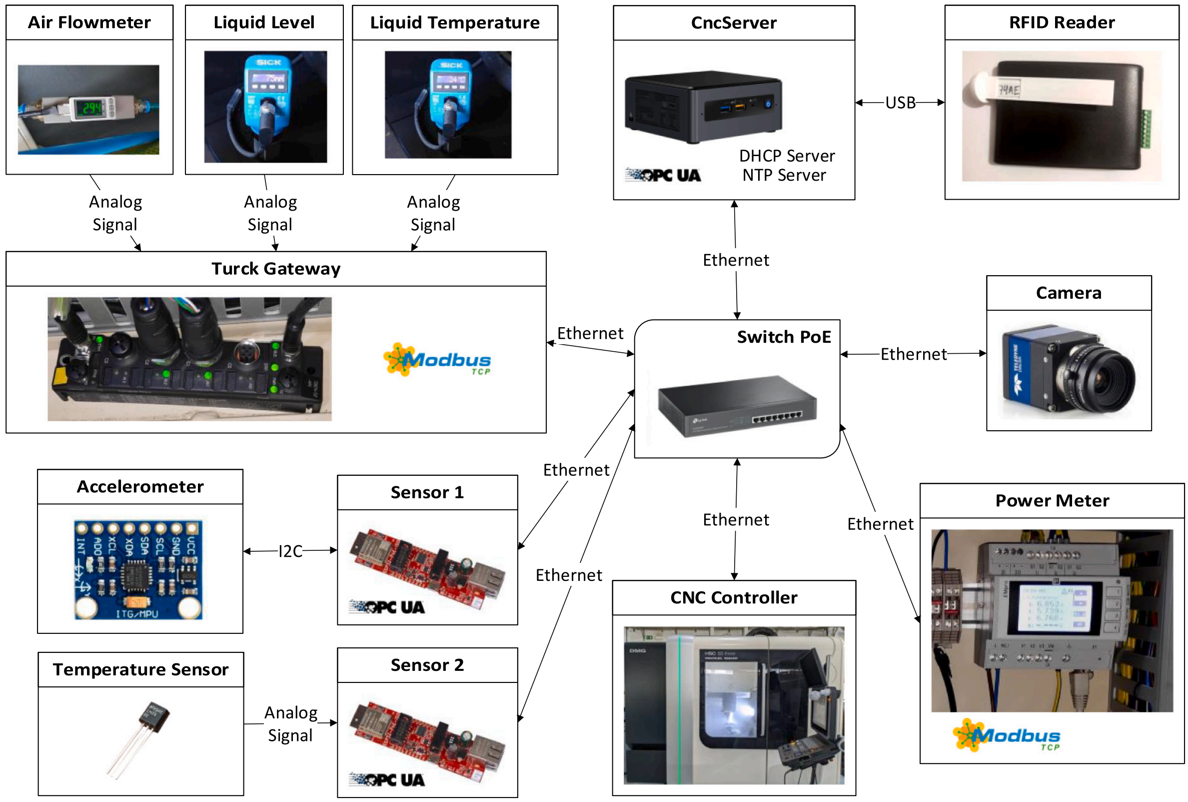

Fig. 7. Example system's diagram for a CNC machine.

A protective case was built, using additive manufacturing technology, to protect the electronics and to allow the installation on a DIN rail in the machine's control cabinet.

Despite the use of LM35 and MPU6050 sensors, this system was developed so that other sensors, which can be read using an Analog-to-Digital Converter (ADC), I2C interface, or any other protocol type available in the ESP32, could be used (note that the use of the ADC could imply the use of a scale adaptation signal conditioning circuit, depending on the sensor-specific output). Additionally, the OPC UA server's information model for exposing a new sensor must be tailored to the sensor's specific characteristics.

The ESP-IDF environment must be configured in order to explore this project and begin changing it to meet the project needs, enabling project creation, building, and flashing to the ESP32, as well as application debugging. The instructions for setting up this development environment can be found in the Espressif official website.[32]

5. Tests and results

The developed system was tested with three different CNC machines with controllers from Fanuc, Heidenhain, and Eding, both in the laboratory and in an industrial environment. Several results were obtained by applying the developed methodologies to the case studies described above, which will be described next. The following equipment was used when conducting these experiments:

- Intel NUC (or another small form factor computer) running Windows 10 Home, with at least one Ethernet port and, for configuration, some peripherals, namely a mouse, a keyboard and a screen;
- PoE switch with at least 6 ports and Ethernet cables;
- Fanuc, Heidenhain and/or Eding CNC controller (or the corresponding simulator version);
- Power meter EEM-MA370 from Phoenix Contact;
- Industrial camera from DALSA Teledyne or other cameras supporting RTSP;
- UHF RFID reader CF-RU5202 from Chafon;
- Turck TBEN-S2-4AI gateway with 4 analog inputs, supporting Modbus TCP/IP communication;
- ESP32-POE-ISO Industrial version from Olimex;
- Additional sensorization as needed, such as temperature sensors, accelerometers, air pressure sensor, among others, as long as supplied with a signal or communication interface that can be read on the ESP32 development board.

Fig. 7, represents the network configuration for each CNC that was tested, with the main difference between each test being the CNC controller in use, according with the case studies described previously.

Concerning Fanuc, the initial tests were conducted in a laboratory environment, using the NCGuide[33] CNC simulator. Because some FOCAS' functions are not available in the simulator, the second stage of the tests was conducted in an industrial environment, using the CNCMonitor application as a datalogger (discussed in Section 3.1). A CNC milling machine equipped with a Fanuc 31i-A controller was used to collect data for two months. It should be noted that switching from the simulator to the real machine required only changing the controller's IP address and the model, in the configuration file, and the application worked just as well with the real machine as it did with the simulator. The hierarchically structured address space of the OPC UA server, associated with the Fanuc CNC machine, is shown in Fig. 2 as an example of the information model. Server address spaces can be accessed using appropriate OPC UA generic clients, such as United

[32] https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/

[33] https://www.fanucamerica.com/products/cnc/software/cnc-guide

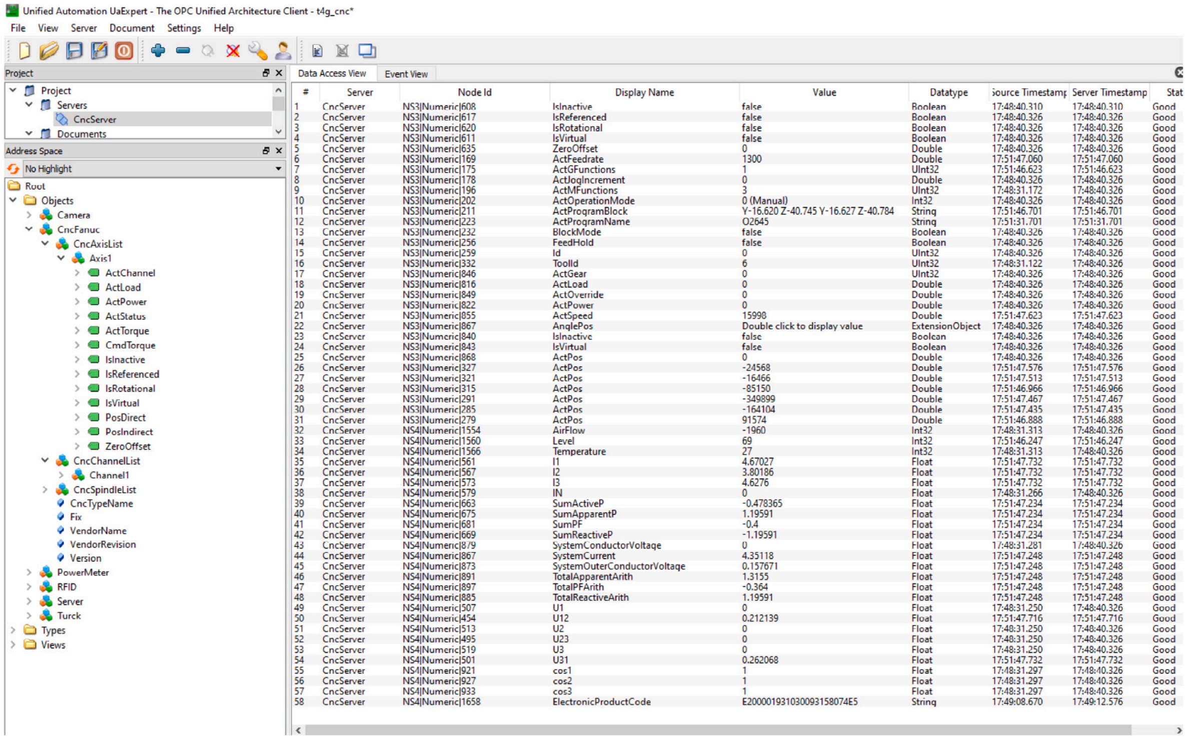

Fig. 8. Screenshot of a UAExpert view of part of the address space variables in real-time.

Automation UAExpert,[34] which was used in this example. Fig. 8 shows a screenshot view of part of the address space variables obtained in real-time using UAExpert.

From the system security perspective, no security was used during the initial development stage. In a later stage, the SignAndEncrypt mode (described in Section 2.3) was activated, providing the authenticity, confidentiality and integrity required when exchanging data, especially in industrial scenarios.

Because the retrofitting process was not yet complete, tests with the Eding CNC controller were limited to their CNC simulator. However, since the Eding CNC simulator and the real CNC controller board run the same software, no differences between simulation and real CNC operation are expected.

The majority of the tests performed with the Heidenhain CNC controller were done with the iTNC530 CNC simulator, with the main limitation of the trial version (which was used) being a maximum of 100 lines of machine instructions. Due to this limitation, the CNC axis was manually moved in the test, with the changes in the cutter position being reflected in the OPC UA server variables. Additionally, preliminary connection and data acquisition tests were performed with a real iTNC530 CNC controller, with the developed application exhibiting the same behavior and results as with the simulator.

A video obtained during some of these tests is available at the following link: https://youtu.be/om7-kGsoIA4.

6. Conclusions and future work

The main goal of this work was to create a standard methodology to develop monitoring solutions for CNC machines, allowing its integration into the Smart Factory environment using OPC UA technology. The developed OPC UA Server provides a standardized approach for data exchange between CNC machines and various OPC UA clients, allowing device control and monitoring while in operation, as well as data transmission to higher-level management systems. The integration of an alarm associated history video feed from the operation, together with the complete history of relevant machining operation variables, allows operators and managers to more easily debug and assess problems and operations status.

In future developments, it will be interesting to explore the data bindings feature of the ASMD software (leveraging machine to sensor connectivity), coupling the Address Space variables with the process sensors, actuators, among other features. As part of future work is also the development of data mining and machine learning techniques, to detect faults and predict events on CNC machines using the gathered data.

Given the wide range of monitoring systems and communication protocols, multi-vendor solutions must take an approach where the various systems available on the market are combined into a single high-level system, allowing data to be collected from a broader range of machining equipment. This is demonstrated in the presented case study, which shows how this can be accomplished by combining leading machining equipment from molds industry companies, in an all-in-one solution.

The interconnection of the Manufacturing Execution System (MES) and the Enterprise Resource Planning (ERP) software with shop floor data supplied from the implemented OPC UA Servers is being developed within the scope of this research, with Key Performance Indicators (KPI) and Overall Equipment Effectiveness (OEE) metrics being used to determine the efficiency and productivity of the machining processes. In this case, the single OPC UA-based server interface is critical in ensuring a consistent communication and data model for CNC machines.

The OPC Foundation, the German machine tool builders association (VDW), and other industrial partners recently formed the Universal Machine Tool Interface (UMATI)[35] working group to develop the OPC

[34] https://www.unified-automation.com/products/development-tools/uaexpert.html

[35] https://opcfoundation.org/markets-collaboration/umati/

UA Companion Specification for Machine Tools. The goal is to create an OPC UA information model that can be used to connect machine tools to ''external'' communication partners such as MES, ERP, automation systems, or the cloud, for which the developed methodology, described in this paper, is well-suited. By adapting the developed interface with the resulting developments of the UMATI working group, it is ensured that the systems based on developed solution will work and support the integration, side-by-side, with future products that support this yet-to-be-defined standard.

CRediT authorship contribution statement

André Martins: Software, Investigation, Validation, Original paper draft, Methodology, Resources. **João Lucas:** Contributed to the original paper draft, Methodology, Resources. **Hugo Costelha:** Conceptualization, Funding acquisition, Supervision, Paper review and editing, Methodology, Resources. **Carlos Neves:** Project administration, Methodology, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] R.Y. Zhong, X. Xu, E. Klotz, S.T. Newman, Intelligent manufacturing in the context of industry 4.0: A review, Engineering (2017) http://dx.doi.org/10.1016/J.ENG.2017.05.015.

[2] R. Giessbauer, E. Lübben, S. Schrauf, S. Pillsbury, Global Digital Operations Study 2018 - How Industry Leaders Build Integrated Operations Ecosystems to Deliver End-To-End Customer Solutions, PwC, 2018.

[3] I. Nicholas, M. Warner, G. Hartmann, A. Sorge, Automating the shop floor: Applications of CNC in manufacturing in Great Britain and West Germany1, J. Gen. Manag. 8 (3) (1983) 26–38, http://dx.doi.org/10.1177/030630708300800302.

[4] J. Downey, S. Bombiński, M. Nejman, K. Jemielniak, Automatic multiple sensor data acquisition system in a real-time production environment, Procedia CIRP 33 (2015) 215–220, http://dx.doi.org/10.1016/j.procir.2015.06.039.

[5] J. Downey, et al., Real time monitoring of the CNC process in a production environment- The data collection & analysis phase, Procedia CIRP 41 (2016) 920–926, http://dx.doi.org/10.1016/j.procir.2015.12.008.

[6] R.T. René de Jesús, H.R. Gilberto, T.V. Iv'n, J.C.J. Carlos, Driver current analysis for sensorless tool breakage monitoring of CNC milling machines, Int. J. Mach. Tools Manuf. 43 (15) (2003) http://dx.doi.org/10.1016/j.ijmachtools.2003.08.004.

[7] P. Stavropoulos, A. Papacharalampopoulos, E. Vasiliadis, G. Chryssolouris, Tool wear predictability estimation in milling based on multi-sensorial data, Int. J. Adv. Manuf. Technol. 82 (1–4) (2016) http://dx.doi.org/10.1007/s00170-015-7317-6.

[8] B. Siddhartha, A.P. Chavan, G.K. HD, K.N. Subramanya, IoT enabled real-time availability and condition monitoring of CNC machines, in: 2020 IEEE International Conference on Internet of Things and Intelligence System, IoTaIS, 2021, pp. 78–84, http://dx.doi.org/10.1109/IoTaIS50849.2021.9359698.

[9] D. Mourtzis, N. Milas, N. Athinaios, Towards machine shop 4.0: A general machine model for CNC machine-tools through OPC-UA, Procedia CIRP 78 (2018) 301–306, http://dx.doi.org/10.1016/j.procir.2018.09.045.

[10] G.M. Martinov, P.A. Nikishechkin, A.A. Khoury, A. Issa, Control and remote monitoring of the vertical machining center by using the OPC UA protocol, IOP Conf. Ser.: Mater. Sci. Eng. 919 (2020) 032030, http://dx.doi.org/10.1088/1757-899X/919/3/032030.

[11] C. Liu, H. Vengayil, Y. Lu, X. Xu, A cyber-physical machine tools platform using OPC UA and MTConnect, J. Manuf. Syst. 51 (2019) 61–74, http://dx.doi.org/10.1016/j.jmsy.2019.04.006.

[12] VDW, OPC Foundation, 'OPC UA Information Model for CNC Systems, Version 1.0, OPC Foundation, 2017.

[13] P.F.S. De Melo, E.P. Godoy, Controller interface for industry 4.0 based on RAMI 4.0 and OPC UA, in: 2019 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2019 - Proceedings, 2019, pp. 229–234, http://dx.doi.org/10.1109/METROI4.2019.8792837.

[14] P. Hunkar, OPC UA vs OPC Classic, DSInteroperability, 2014.

[15] S. Cavalieri, M.G. Salafia, M.S. Scroppo, Integrating OPC UA with web technologies to enhance interoperability, Comput. Stand. Interfaces 61 (2019) 45–64, http://dx.doi.org/10.1016/j.csi.2018.04.004.

[16] B. für Sicherheit in der Informaationstechnik, OPC UA Security Analysis'ô, Für Sicherheit in Der Informaationstechnik, Bundesamt, 2017.

[17] C.V. Neu, I. Schiering, A. Zorzo, Simulating and detecting attacks of untrusted clients in OPC UA networks, in: ACM International Conference Proceeding Series, New York, New York, USA, 2019, pp. 1–6, http://dx.doi.org/10.1145/3360664.3360675.

[18] J. Imtiaz, J. Jasperneite, Scalability of OPC-UA down to the chip level enables 'Internet of Things', in: IEEE International Conference on Industrial Informatics, INDIN, 2013, pp. 500–505, http://dx.doi.org/10.1109/INDIN.2013.6622935.

[19] VDMA and fraunhofer IOSB-INA, in: Industrie 4.0 Communication Guideline Based on OPC UA'Ô, VDMA, Fraunhofer IOSB-INA, 2017.

[20] P. Drahos, E. Kucera, O. Haffner, I. Klimo, Trends in industrial communication and OPC UA, in: Proceedings of the 29th International Conference on Cybernetics and Informatics, K and I 2018, abril 2018, vol. 2018-January, 2018, pp. 1–5, http://dx.doi.org/10.1109/CYBERI.2018.8337560.

[21] OPC Foundation, OPC UA Specification Part 14 - PubSub 1.04, OPC Foundation, 2018.

[22] R. Zhao, R. Yan, J. Wang, K. Mao, Learning to monitor machine health with convolutional bi-directional LSTM networks, Sensors 17 (2) (2017) 2, http://dx.doi.org/10.3390/s17020273.

[23] A. Shukla, Y. Pansuriya, S. Tanwar, N. Kumar, Md. J. Piran, Digital twin-based prediction for CNC machines inspection using blockchain for industry 4.0, in: ICC 2021 - IEEE International Conference on Communications, 2021, pp. 1–6, http://dx.doi.org/10.1109/ICC42927.2021.9500498.

[24] Z.M. Çinar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, B. Safaei, Machine learning in predictive maintenance towards sustainable smart manufacturing in Industry 4.0, Sustainability 12 (19) (2020) 19, http://dx.doi.org/10.3390/su12198211.

[25] N.R. Tambake, B.B. Deshmukh, A.D. Patange, Data driven cutting tool fault diagnosis system using machine learning approach: A review, J. Phys.: Conf. Ser. 1969 (1) (2021) 012049, http://dx.doi.org/10.1088/1742-6596/1969/1/012049.

[26] S. Profanter, A. Perzylo, M. Rickert, A. Knoll, A generic plug & produce system composed of semantic OPC UA skills, IEEE Open J. Ind. Electron. Soc. 2 (2021) 128–141, http://dx.doi.org/10.1109/OJIES.2021.3055461.

[27] L. Underberg, R. Kays, S. Dietrich, G. Fohler, Towards hybrid wired-wireless networks in industrial applications, in: Proceedings - 2018 IEEE Industrial Cyber-Physical Systems, ICPS 2018, 2018, pp. 768–773, http://dx.doi.org/10.1109/ICPHYS.2018.8390804.

[28] G. Cena, A. Valenzano, S. Vitturi, Hybrid wired/wireless networks for real-time communications, IEEE Ind. Electron. Mag. 2 (1) (2008) 8–20, http://dx.doi.org/10.1109/MIE.2008.917155.

[29] Fanuc and Inventcom, FANUC Open CNC API specifications, 2018, Inventcom.

[30] S.H. Atluru, A. Deshpande, Data to information: can mtconnect deliver the promise, in: 37th Annual North American Manufacturing Research Conference, NAMRC 37, 2009.

[31] P. Smid, Fanuc CNC Custom Macros: Programming Resources for Fanuc Custom Macro B Users, Industrial Press, 2005.

[32] J. Lucas, Desenvolvimento de Servidor OPC UA Para Sistema CNC, (Master Thesis), Leiria, 2019, [Online]. Available: http://hdl.handle.net/10400.8/4553.

[33] Espressif, Esp32-WROOM-32 datasheet, 2019, Espressif.