Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 53 (2016) 198 – 205

www.elsevier.com/locate/procedia

The 10th International Conference on Axiomatic Design, ICAD 2016

Axiomatic design applied to the development of a system for monitoring and teleoperation of a cnc machine through the internet

Oliveira, L.E.S.[a], Álvares, A.J.[a]

[a]*Automation and Control Group, Mechanical Engineering Department, University of Brasilia, Brasilia, CEP 70910900, Brazil*

[*] Corresponding author. Tel.: +55-061-8177-2668. *E-mail address:* oliveira.luiz@aluno.unb.br

---

Abstract

The new era of manufacturing, called industry 4.0, will require the intensive use of mechatronics products. In this context, there are Cyber-Physical Systems and Internet of Things (IoT). These are represented by platforms that are integrated through connectivity protocols that permit a wide sharing of information among different devices. The design of these applications requires a methodology that sets requirements by reducing the complexity inherent in the development of these systems. As such, the Axiomatic Design (AD) has the capacity to simplify this kind of design process. This work studies the application of Axiomatic Design to the planning of a system for monitoring and teleoperation through the Internet of a Romi-Galaxy 15M machine tool with CNC Fanuc 18i-Ta, using the open standards OPC-UA and MTConnect. As a result, the functional requirements (FR) were formulated and, consequently, the design parameters (DP), which are represented by classes in an object-oriented application.

Peer-review under responsibility of the scientific committee of The 10th International Conference on Axiomatic Design

Keywords: Axiomatic design; industry 4.0; cnc; mtconnect

---

1. Introduction

Software systems have been playing an important role in manufacturing for a few decades. In this context, they are able to monitor and control a series of processes, which can be a machine located on the shop-floor, or the organizational management-level activities. Recently, information technology systems are drawing increasing attention within the corporate and product strategy of manufacturing companies along with the rise in Germany of the paradigm of the fourth industrial revolution, also called Industry 4.0.

The Industry 4.0 paradigm has as main purpose the consolidation of smart factories to allow the creation of on-demand, mappable and customized products, in order to meet the individual needs of customers [1]. This new era of industry is based on Cyber-physical systems (CPS) and Internet of things (IoT) technologies, using the power of information and communication technologies (ICTs) integrated into the shop-floor. In other words, this new paradigm promotes a virtualized world in the context of manufacturing, converting physical systems in services, using ICTs extensively.

CPSs represent the joining of embedded systems, the Internet, data and services available online [2]. These systems are focused on a process-integrated network and physical resources that allow the fusion of real and virtual systems [3]. In accordance with Kagermann et al. [4] the Internet of things and services (Fig. 1) enables creating networks incorporating the entire manufacturing process that converts factories into a smart environment. For Gubbi et al. [5], this is achieved through the presence of attributes such as integrated ubiquitous sensing, analysis of data and information representation with cloud computing as a unified infrastructure.

The Internet of things is a network where sensors and actuators can communicate through protocols, be identified, located, tracked, monitored and managed intelligently. In this context, these devices can share information between platforms through a unified framework [5].

The implementation of smart factories will be carried out gradually by the companies and involves features that should be treated as requirements, such as: horizontal integration; end-to-end engineering; vertical integration and networked manufacturing systems; end-to-end transparency; a comprehensive broadband infrastructure for industry; energy efficiency; and standardization and reference architecture. This last feature involves the adoption of key standards. Regarding the capacity to connect manufacturing equipment through Web-based networks, a well-developed standard is the MTConnect [6]. It is royalty-free and was developed with the purpose to become a

Peer-review under responsibility of the scientific committee of The 10th International Conference on Axiomatic Design
doi:10.1016/j.procir.2016.06.099

standard to promote the integration of CNC machines, being based on the most important existing standards in the field of manufacturing and the software industry, such as HTTP (Hypertext Transfer Protocol) and XML (Extensible Markup Language) [7].

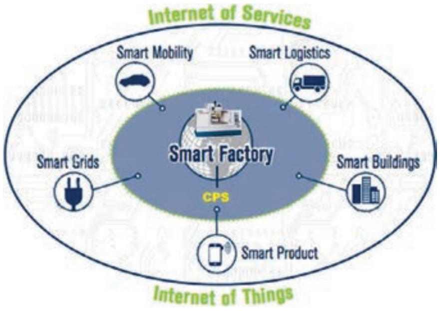


Fig. 1. Industry 4.0 and smart factories as part of the Internet of Things and Services [4]

A typical architecture using the MTConnect standard consists of two software elements: Adapter and Agent. The first has the function of acting as a "translator" of the data structure of the CNC machines in a standardized format, and the second one acts as a buffer to store the data from the adapter and distribute them in XML format to the client applications (for example, web-based supervisory systems). Figure 2 illustrates the data flow between the elements of the monitoring system architecture based on MTConnect (Cyber-Physical Production Systems).

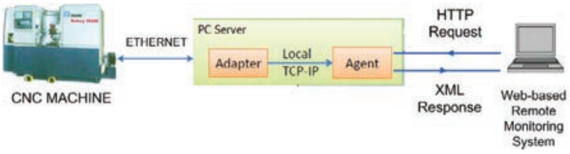


Fig. 2. Information flow among MTConnect elements (CPS)

Another important standard to promote remote connectivity in the manufacturing environment is the OPC-UA (Object linking and embedding for process control united architecture). This standard was developed by the OPC Foundation and emerged as an evolution of the classic OPC to represent a platform-independent interoperability standard in the exchange of data between the manufacturing shop-floor and the enterprise [8]. While MTConnect facilitates the connection of machine tools and other manufacturing equipment linked with a network for gathering data, OPC-UA promotes the needed interoperability for data communication throughout the plant [6].

The evolution of industry 4.0 happens essentially through the development and implementation of software systems and connectivity protocols aimed at integration and automation of the manufacturing environment. In this context of technological innovation, the early stages of a software design are surrounded by many uncertainties and lack of accurate information. In this kind of situation, the Axiomatic Design methodology (AD) is quite effective in the early stages of the design, when the deployment of customer needs in the design requirements (called Functional Requirements - FRs) is performed, and through these the design parameters (DPs) are extracted.

Suh [9,10] held a robust work related to software system design based on Axiomatic Design theory and presents the methodology as a platform applicable to all types of systems including those consisting only of software and those that involve hardware and software.

The combination of axiomatic design and object-oriented technologies resulted in a generic approach called ADo-oSS (Axiomatic Design of Object-Oriented Software Systems) [9, 10]. This methodology makes software development less subjective and provides a more practical and rational resource to this activity, helping to reduce or eliminate the need for debugging and excessive changes in design. One of the final outputs of the ADo-oSS is the system architecture, which is represented by a flow chart, which aids software developers.

Thus, this work presents the development of a system architecture (definition of FRs and DPs) for monitoring and teleoperation of a CNC machine tool through the Internet based on the combination of the features of object-oriented methodology, currently widely used in the development of software systems, through programming languages such as Java and C++, and axiomatic design. The combination of these methodologies is also used to contextualize the designed system as part of systems for industry 4.0. In this sense, the system design has as one of its main constraints the combined use of MTConnect and OPC-UA standards.

2. Axiomatic design for software systems development

In the beginning, Axiomatic Design represented a powerful set of guidelines to aid the development of optimized designs of hardware systems. Originally applied to the design of mechanical assemblies, the increase in the perception of the power of Axiomatic Design allowed the inclusion of other disciplines such as engineering and manufacturing systems [11]. This led to an increasing interest in the application of AD to software design since the 1990s.

A pioneering work on the software design based on axiomatic design was conducted by Kim, Suh and Kim [12]. In that work the authors provide the scientific basis for software design, reacting to the lack of a methodology that had a set of fundamental principles and with criteria for decision-making.

The conjunction among the object-oriented methods and axiomatic design enabled the generation of a generic approach to the design of a software called ADo-oSS (Axiomatic Design of Object-Oriented Software Systems) [9,10,13]. This is structured under the V-model and combines the power of axiomatic design with the popular object-oriented programming techniques, and emerged as a way to overcome the shortcomings of existing techniques that support software design, such as extensive debugging and testing. This methodology has as final output the system architecture, represented by the flowchart. The generated flowchart can be applied to different purposes, such as guiding the construction of the data model (Fig. 3), aiding in the management of design tasks, and in the improvement of design proposal through the identification of a coupled design. ADOs-oSS is implemented in systems involving only software, like the Acclaro software [10,13], and also in systems combining hardware/software [9].

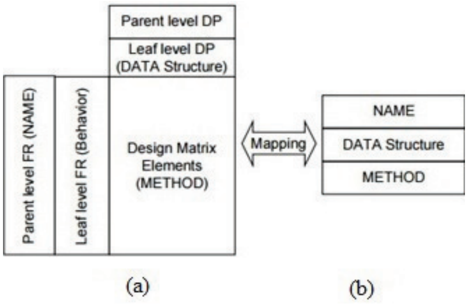


Fig. 3. The correspondence between the full design matrix and the OOT diagram, (a) complete design matrix table, (b) Class diagram [13]

Pimentel and Stadzisz [14] also have a use case of axiomatic design theory for the design of object-oriented software. In this approach, AD is integrated with UML and the unified software development process with the purpose of ensuring quality design solutions throughout the development process. Clapis and Hintersteiner [15] combine the capacity to generate functional requirements documentation and traceability between FRs and design of objects of axiomatic design software with the OMT (Object-Modeling Technique) methodology, to increase the value of the software design process.

In designs involving hardware and software, Hintersteiner and Nain [16] proposed a methodology based on the axiomatic design to facilitate software design of a control system in conjunction with its corresponding hardware system.

Based on the V-model for the Axiomatic Design Process for Object-oriented Software systems [9,10,13,17], Park and Yi [18] developed a software through the concept of a sequential algorithm using orthogonal matrix for optimization of designs in discrete space, whereas the optimization is developed in continues space.

The example of a digital clock design helped to demonstrate the complementarity between AD and sequence enumeration [19]. In this work, the sequence is used to identify DPs as operating states of the clock based on stimuli and responses. AD focuses on high-level analysis whereas the sequence enumeration focuses on the low-level analysis.

Emphasizing object-oriented programming, the Design Structure Matrix (DSM) and the Multiple Domain Matrix (MDM) complemented the axiomatic design matrix (DM) in the tasks of defining, modeling, and managing the complexity within the classes (intra-classes complexity) [20]. Do [21] proposed an extension to the axiomatic process, in addition to the process of decomposition and mapping between the design domains. New domains were associated with the symmetric domains (FR, DP, and PV).

Togay et al. [22] extended the axiomatic design in a rule-based approach to guide stakeholders (customers, developers, etc.) during design decisions and selection of systems' features. In the design hierarchy, both functional requirements (FRs) and design parameters (DPs) are decomposed through rules based on the following classification: optional, mandatory, alternative and OR. El-Haik and Shaout [23] feature AD as an engine to generate software conceptual alternatives incorporated to the methodology DFSS (Design for Six Sigma), developed by those authors.

The AD methodology was also used in usability design for user interface representing a rational design process, useful in the task of capture and communication of the knowledge of previous designs [24] and the reduction or elimination of coupling in usability analysis of Human-Computer Interaction [25]. Schreyer and Tseng [26,27] studied the use of AD to support the design of PLC (Programmable Logic Controller) software.

3. Applying axiomatic design for development of a software system in compliance with Industry 4.0

The implementation of the axiomatic design in the developed architecture had as main guidelines the V-model, which addresses the AD process for object-oriented software systems, and the ADo-oSS approach [9,10].

3.1. Definition of FRs and mapping between the domains

The first stage of implementation of the methodology involves the definition of requirements from potential customers (customers' needs, CNs) in the customer domain, involving mainly manufacturing companies and CNC machine tools providers engaged in the development of the Smart Factories. Those needs have become mapped when considering the technological challenges of these companies to develop products in the context of industry 4.0. In addition, for selecting the main customers' needs, some of the attributes that include cyber-physical and internet of things systems that will be part of the proposed system architecture are the following: adoption of standardization and reference architecture; vertical integration and networked manufacturing systems; and end-to-end transparency. This first stage resulted in Table 1.

Table 1. Costumers Needs

Costumers Needs	Description
CN1	View the current state of the process
CN2	View the current state of the machine
CN3	Allow to check machine operation historical data
CN4	Warn when there is a problem in the process
CN5	Provide a view of the factory floor in real time
CN6	Monitor the tool Status
CN7	Intervene in the manufacturing process remotely
CN8	Obtain data about the part
CN9	User-friendly interface
CN10	Display charts
CN11	Provide reports

The relatively large number of identified customer requirements represents the complexity of the design under analysis, since this is a monitoring and operation system architecture with more than one layer of software and has compliance with Industry 4.0.

The next step was the choice of design constraints:

Cs1: the system is based on the Internet as a service;

Cs2: Use key-standards in compliance with the new industrial revolution;

Cs3: having the data source available in the cloud.

Cs4: Virtual monitoring and control.

Cs5: Platform independent system.

After defining the constraints, it was determined the functional requirement to zero level (FR0), representing the main requirement of the hierarchy, and the corresponding design parameter (DP0):

FR0: To Build a tool for remote operation and monitoring of a CNC machine-tool, compliant with Industry 4.0 features.

DP0: A Web-based System for Teleoperation and Monitoring of a CNC machine-tool (CAM).

Moving to the next level, the main functional requirement was decomposed in the following sub-requirements:

FR1: Monitor the manufacturing process remotely.

FR2: Interactivity with the process and the machine.

FR3: Create an intuitive user interface.

FR4: Define mechanisms for accessing historical data of the machine operation.

With the axiomatic design strategy adopted in this work, the functional requirements (FRs) above have been mapped to the following design parameters (DPs):

DP1: Functions for Web-based process monitoring.

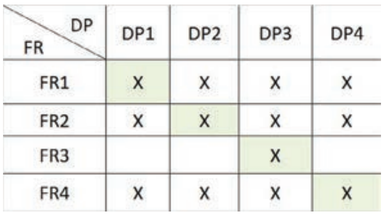
DP2: Functions for machine configuration and control through the Internet.

DP3: Structured Graphical User Interface (GUI).

DP4: System Query integrated with the machine operation database.

The mapping between the functional and physical domains in this first level resulted in the matrix shown in Fig.4.

The result of the first-level mapping reveals a design significantly coupled. This fact leads the design process to the second level of decomposition. The decomposition of FR1, FR2, FR3 and FR4 resulted in a total of 45 FRs for the second level. With the purpose of illustrating this design process step, Table 2 lists the FRs and DPs in the mapping of the branches FR1 and DP1 in the design hierarchy.



DP FR	DP1	DP2	DP3	DP4
FR1	X	X	X	X
FR2	X	X	X	X
FR3			X	
FR4	X	X	X	X

Fig. 4. Mapping for the first level

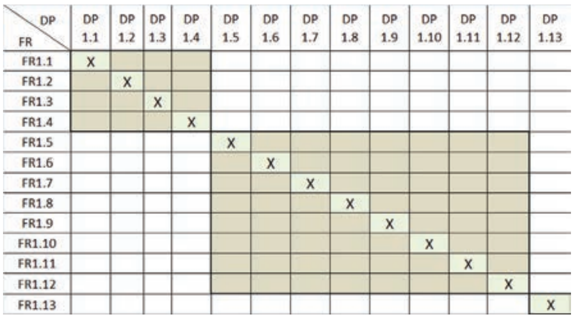
Fig.5 shows the design matrix as a sequence of the mapping made in Table 2.

The mapping process of the FR1.x level reveals a tendency towards decoupling the design considering the modules related to the functional requirement ”Monitor the manufacturing Process remotely”. The decomposition of FR1 separated the modules into OPC Server objects (FR1.1-FR1.4) and MTConnect Adapter classes (FR1.5-FR1.12), which are integrated to monitor the CNC turning parameters. In addition, there is the video/audio server to monitor the manufacturing cell through images and sound, WebCam (FR1.13) [28], previously developed in the Graco (Group of automation and control) facilities, at the University of Brasilia (UnB).

Mapping is made for the leaf level to the FR2.x branch of the design hierarchy (Table 3), which is represented by the FRs and DPs from the second level decomposition in the matrix of Fig.6.

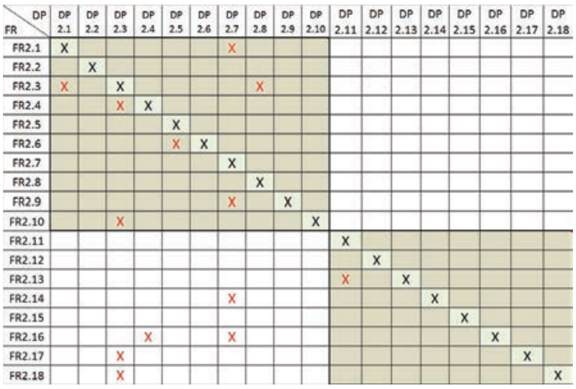
Table 2. Second-level mapping for FR1.x in DP1.x

x	FR1.x	DP1.x
1	Acquire Power machine Status(On/Off)	OPC object to acquire
2	Acquire the control mode	OPC object to acquire the variable Control Mode
3	Acquire machine door status	OPC object to acquire the status of the machine door
4	Acquire cooling status	OPC object to acquire the status of the coolant system
5	Acquire current feed-rate of the axes	method getActualFeedRate
6	Acquire axes’ current position	method getAxesPosition (for Axes X/Z/C)
7	Acquire the axes’ load	method getAxesLoad (for Axes X/Z/C)
8	Acquire Alarms Status	method getAlarmDescription
9	Acquire the name of the running program	method getCurrentProgram
10	Acquire axes’ rotation (Spindle Speed)	method getSpindleSpeed
11	Acquire travel distance for the spindles	method getDistanceToGo
12	Acquire configuration data	method getDataSettings
13	Acquire images and audio from the shop-floor	WebCam Server (Graco/UnB)



DP FR	DP 1.1	DP 1.2	DP 1.3	DP 1.4	DP 1.5	DP 1.6	DP 1.7	DP 1.8	DP 1.9	DP 1.10	DP 1.11	DP 1.12	DP 1.13
FR1.1	X												
FR1.2		X											
FR1.3			X										
FR1.4				X									
FR1.5					X								
FR1.6						X							
FR1.7							X						
FR1.8								X					
FR1.9									X				
FR1.10										X			
FR1.11											X		
FR1.12												X	
FR1.13													X

Fig. 5. Second level decomposition for FR1



DP FR	DP 2.1	DP 2.2	DP 2.3	DP 2.4	DP 2.5	DP 2.6	DP 2.7	DP 2.8	DP 2.9	DP 2.10	DP 2.11	DP 2.12	DP 2.13	DP 2.14	DP 2.15	DP 2.16	DP 2.17	DP 2.18
FR2.1	X						X											
FR2.2		X																
FR2.3	X		X					X										
FR2.4			X	X														
FR2.5					X													
FR2.6					X	X												
FR2.7							X											
FR2.8								X										
FR2.9							X		X									
FR2.10			X							X								
FR2.11											X							
FR2.12												X						
FR2.13											X		X					
FR2.14							X							X				
FR2.15															X			
FR2.16				X			X									X		
FR2.17			X														X	
FR2.18			X															X

Fig. 6. Second level decomposition for FR2

After mapping the FRs in DPs for the second level, it is possible to build the full design matrix since the data sources and parameters, that meet the listed functional requirements, were identified.

Table 3. Second-level mapping for FR2.x in DP2.x

x	FR2.x	DP2.x
1	Transmit a NC program to the machine controller	OPC object to transmit an NC Program to the controller
2	Receive a NC program from the machine controller	OPC object to receive an NC program from the controller
3	Start program execution (Cycle Start)	OPC object to start an NC program execution
4	Stop or Cancel program execution (Cycle Stop/RESET)	OPC object to cancel/stop the execution of an NC program
5	Turn On Refrigeration (Coolant ON)	OPC object to turn on the machine coolant
6	Turn Off Refrigeration (Coolant OFF)	OPC object to turn off the machine coolant
7	Define the controller operation mode	OPC object to activate/deactivate controller operation modes (AUTO / EDIT / MDI /JOG)
8	Activate/Deactivate program execution options	OPC object to activate/deactivate program execution options (SINGLE BLOCK / BLOCK DEL/ PROG TEST/ OPT STOP / DRY RUN)
9	Move axes manually out of operation	OPC object to handle directional keys of axes (+ X / -X / + Z / -Z)
10	Show tool data	OPC object to retrieve tool status information
11	Search a specified program	method searchProgramNC
12	Delete a specified program	method deleteProgNC
13	Open the CNC program directory	method openProgramDir
14	Activate zero-offset value	method sendWorkZeroOffSet
15	Write default axis relative position	method setRelAxisPosition
16	Transmit command lines to CNC	Method sendMDIProgCommand
17	View program block execution	method getProgrblock
18	Receive error messages	Method getErrorMessages

3.2. *Definition of the full-design matrix*

Functional requirements related to ”Monitor the manufacturing Process remotely” were met through the machine teleoperation server represented by the OPC-UA Server, MTConnect Server (adapter and Agent) and the WebCam server for providing images and audio from the shop-floor. The first four FRs (FR1.1-FR1.4) from the matrix of Fig.5 have as inputs (DPs) PMC (CLP) parameters from the lathe Romi Galaxy 15M’s controller (Fanuc 18i-Ta), that are provided through the API Focas 2 (Fanuc Open CNC API Specifications 2) for remote access to these functions. These variables are monitored through the data access service of the OPC-UA server. The CNC parameters (DP1.5-DP1.12, Fig.5) are made available to the client application through data transmission requests issued from the adapter to the MTConnect Agent located in the cloud.

The modules related to the requirements of ”Interactivity with the process and the machine” are represented in the matrix of Fig.6. In this matrix there are inputs that will be provided by the PMC(CLP) functions of the machine controller, such as DP 2.3: ”OPC object to start the NC program execution” (FR 2.3: Cycle Start); or DP 2.7: ”OPC Object to activate/deactivate the controller operation modes (AUTO/EDIT/MDI/JOG)” (FR2.7: Controller Mode). To control these functions, the OPC-UA Server will also be used. From module 2.11 are the main methods corresponding to machine’s control activities, such as ”delete a specified NC program” and ”transmit command lines to CNC” that work by sending command lines.

In order to try to meet the Independence axiom fully and approximate modules with affinities in terms of technological solutions, such as those related to OPC-UA server, the full design matrix was rearranged and the result is shown in Fig.7, complemented by Table 4.

The full design matrix defines the modules which enables mapping the physical domain into the process domain represented by process variables (PV), which in the case of software designs correspond to the real source codes. In this step, objects, attributes, operations and interfaces are identified to compose the client/server system data model.

Table 4. Off-Diagonal Full Matrix elements

Off-Diagonal Element	Description
J	Represents a control sequence
K	Represents a control sequence
L	I/F between OPC-UA Server and client application for issuing commands.
M	OPC-UA Server class to acquire machine status data to the GUI
N	I/F between the adapter and the GUI (MTConnect Agent)
O	I/F between the GUI and the video/audio server (WebCam)
P	I/F Class between the GUI and remote operation server of the CNC machine.
Q	Class for storing and querying operation historical data from OPC-UA Server
R	Class for storing and querying CNC’s historical data from MTConnect Server
S	Class to query alarms database

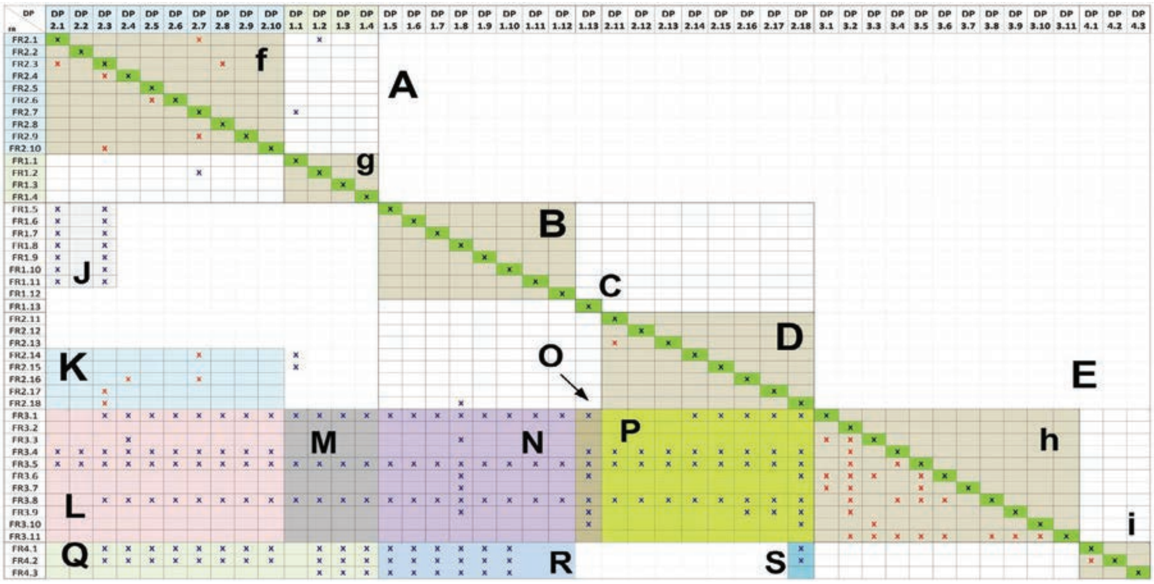


Fig. 7. Full design matrix (Rearranged)

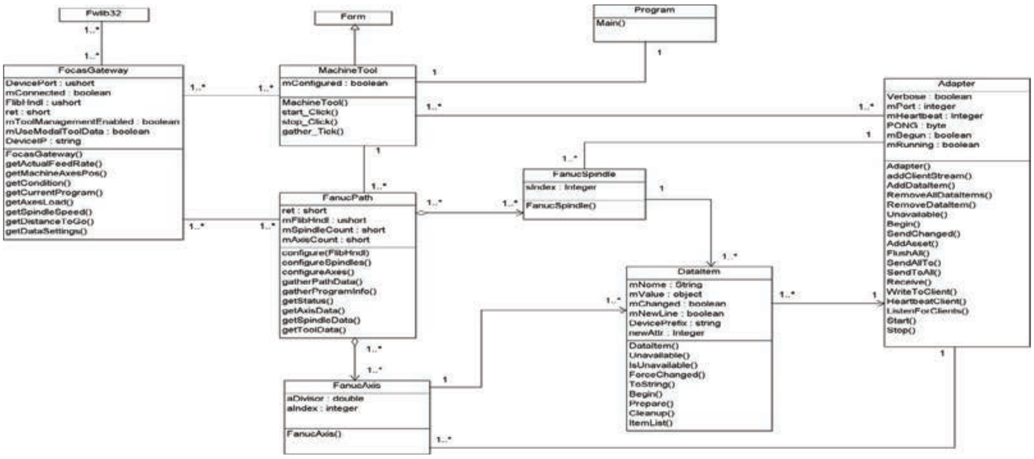


Fig. 8. Class diagram of the Adapter (MTConnect Server)

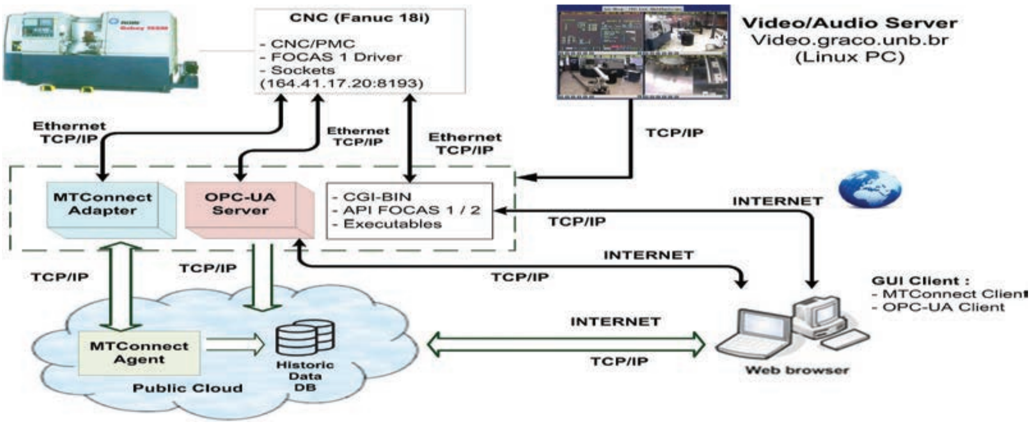


Fig. 9. Cyber-Physical Production Systems: architecture, data flow and some available services

3.3. System architecture generation

The full design matrix enables the identification of the layers of software system architecture. At the matrix upper left corner the elements of (A) OPC-UA Server stand out, formed by their writing classes (f), which control the OPC command tags and the interaction with the machine tool, and reading classes (g), which acquire the status parameters of the machining operation. The element (B) is represented by the adapter methods (MTConnect Server) that encapsulate the DNC functions from the machine controller, which are provided by the Focas 2 protocol library, *fwlib32.dll*, which is written in C language. Module (C) (FR 1.13) represents the video/audio streaming server (WebCam), which is complemented by element (D) to compose the CNC machine teleoperation server. Classes derived from (D) represent an intermediate layer between the client and the CNC that make use of web access mechanisms such as C++ scripts, to issue commands to the CNC machine server.

The classes that perform the user interface (GUI) control are represented by the modules block (E) in the design matrix, which integrates the functional requirements for remote control and process monitoring (h), and historical data query, reporting and charts generation (i).

The off-diagonal terms highlighted in Figure 7 show the inter-relationships between FRs and DPs placed in different branches. These guide the integration sequence between objects and classes. The description of each off-diagonal element is provided by Table 4.

For the purpose of illustrating the result of the implementation of software axiomatic design, Figure 8 shows a portion of the MTConnect adapter class diagram (FR1.5-FR1.12) presenting the class *FocasGateway* with methods that encapsulate the main functions provided by the Focas 2 Protocol library for communication with the CNC.

The results obtained through the adaptation of the AD methodology with object-oriented techniques, as the design matrix and the class diagram, aid to coordinate the implementation tasks of the client/server system architecture.

The analyzed architecture in this work represents a manufacturing support system that is placed in the context of electronic manufacturing (*E-manufacturing*), specifically remote manufacturing systems that use the internet as a communication channel. In this context, works that represent important references in that topic were developed, such as the Webturning system [28] for remote manufacturing of rotational parts, that integrates the WebMachining methodology [29], and WebOxicorte [30], a GUI (Graphical User interface) for web-based teleoperation of a CNC oxi-cutting machine. These methodologies associated with the references of systems for industry 4.0 helped to determine the details that make up the final system architecture, illustrated in Figure 9.

The generated architecture represents a client/server system that has the machine controller (CNC/PMC) as a data server that provides parameters about the turning process into a proprietary format. The data related to the CNC are ordered and sent to the adapter where they are converted to a modified version of SHDR (Simple Hierarchical Data Representation) format and sent to an MTConnect Agent located in the Cloud, where it also runs a database on the manufacturing process, used as historical data. The PMC (CLP) functions of the machine will be monitored and controlled by an OPC client located in the client application, associated with an OPC-UA server. The DNC (Distributed Numerical Control) functions, which are not supported by the MTConnect Protocol, will be enabled through a mechanism of scripts (CGI, for example) to send commands to the controller through the Internet. The whole process, the machine, and the shop-floor are also monitored through an audio/video streaming system (WebCam), built in Java Applets. All information about the machine operation is made available through a Web client accessed through a browser or, in the future, through other internet-based platforms.

4. Conclusion

Axiomatic design is extensively applied to the design and development of object-oriented software systems for a few decades. The AD methodology applied to the design process of object-oriented software systems (ADo-oSS) systematizes the software design process to aid in the management of design tasks, reducing debugging and modification efforts, and increasing the extensionality and reusability of software, making software development a science and not an art.

The use of axiomatic design for the design of a system for remote supervision of a CNC machine, in compliance with the requirements of industry 4.0, demonstrates the methodology's versatility showing that it is possible to implement not just a single software system on a single layer, but also for a client/server software architecture involving several layers.

This work also demonstrated the importance of AD methodology for the development of industry 4.0, including designs to cyber-physical systems (CPS) and internet of things (IoT), promoting the independence axiom between functional requirements, a useful principle for architecture designs consisting of stand-alone modules, but without disregarding integration relationships between those systems.

The bandwidth and the inherent delays of TCP/IP impose a strong constraint to the teleoperation systems through the Internet [28]. To solve this problem it is necessary to endow the teleoperation system, in the server close to the CNC, with mechanisms that enable decision making in critical situations, without depending on the client side, in the case of the user/operator. The capture of images in real time is fundamental to allow immersion of the operator in the system, enabling greater safety for sending commands. As there is an inherent delay to TCP/IP, a lot of care should be taken in the command actions executed remotely.

Therefore, it is necessary to endow the system with some intelligence to solve conflicts that can happen during the teleoperation process. To provide the system with safer teleoperation, an architecture for supervisory control should be used, where the control actions are executed locally, starting with the request by a remote command performed by the client. Thus, the WebTurning system is classified as "supervisory control prevailing the control accomplished by the human operator."

For future works, the axiomatic design methodology could be properly adapted to the design of other systems related to Industry 4.0 technologies, as a full-service architecture for manufacturing available through a completely cloud-based platform.

Acknowledgements

The authors would like to thank the financial support of the Brazilian government agency MEC/CAPES/CNPq.

References

[1] Pisching MA, Junqueira F, Santos Filho DJ, Miyagi PE. An Architecture For Organizing And Locating Services To The Industry 4.0.Proceedings of 23rd ABCM International Congress of Mechanical Engineering 2015. p.8.

[2] MacDougall W. Industrie 4.0: Smart Manufacturing for the future: Germany Trade and Invest; 2014.

[3] Gorecky D, Schmitt M, Loskyll M, Zühlke D. Human-Machine-Interaction in the Industry 4.0 Era. Proceedings of INDIN2014 12th IEEE International Conference on Industrial Informatics 2014; 12:289-94.

[4] Kagermann H, Wahlster W, Helbig J. Recommendations for Implementing the Strategic Initiative Industrie 4.0: Final Report of the Industrie 4.0 Working Group; 2013.

[5] Gubbi J, Buyyab R, Marusica S, Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems 2013; 29:1645–1660.

[6] Albert M. Seven Things to Know about the Internet of Things and Industry 4.0. Modern Machine Shop Magazine; 2015.http://www.mmsonline.com/articles/7-things-to-know-about-the-internet-of-things-and-industry-40.

[7] Sobel W. MTConnect® Standard Part 1: Overview and Protocol. 1.1.0: MTConnect Institute; 2014 http://www.mtconnect.org.

[8] Mahnke W., Leitner SH, Damm, M. OPC Unified Architecture, 1st ed.Berlin: Springer Verlag; 2009.

[9] Suh, NP. Axiomatic Design: Advances and Applications. New York: Oxford University Press; 2001.

[10] Suh NP, Do S. Axiomatic Design of Software Systems. CIRP Annal Manufacturing Technology 2000;49:95-100.

[11] Hintersteiner JD. Addressing Changing Customer Needs by Adapting Design Requirements. Proceedings of ICAD2000 First International Conference on Axiomatic Design 2000;1:290-9

[12] Kim SJ, Suh NP, Kim SK. Design of Software Systems Based on Axiomatic Design. Annals of the CIRP 1991;40:165-70.

[13] Do SH, Suh NP. Object-Oriented Software Design with Axiomatic Design. Proceedings of ICAD2000 First International Conference on Axiomatic Design 2000;1:278-84.

[14] Pimentel AR, Stadzisz PC. A Use Case based Object-Oriented Software Design Approach using The Axiomatic Design Theory. Proceedings of ICAD2006 Fourth International Conference on Axiomatic Design 2006;4:1-8.

[15] Clapis PJ, Hintersteiner JD. Enhancing Object-Oriented Software Development throught Axiomatic Design. Proceedings of ICAD2000 First International Conference on Axiomatic Design 2000;1:272-77.

[16] Hintersteiner JD, Nain AS. Integrating Software into Systems: An Axiomatic Design Approach. Proceedings of the Third International Conference on Engineering Design and Automation 1999;3:1-7.

[17] Do, S.H., Suh, N.P., Systematic OO Programming with Axiomatic Design, Computer 1999;32:121-24.

[18] Yi JW, Park G. Software Development of a Sequential Algorithm with Orthogonal Arrays (SOA) Using Axiomatic Design. Proceedings of ICAD2004 The Third International Conference on Axiomatic Design 2004. p.7.

[19] Woolley B, Li Z, Tate, D. The Application of Sequence Enumeration to the Axiomatic Design Process. Proceedings of ICAD2013 The Seventh International Conference on Axiomatic Design 2013;7:195-99.

[20] Li Z, Tate D. Managing Intra-Class Complexity with Axiomatic Design and Design Structure Matrix Approaches. Proceedings of ICAD2011 The Sixth International Conference on Axiomatic Design 2011;6:142-51.

[21] Do S. Software Product Lifecycle Management Using Axiomatic Approach. Proceedings of ICAD2004 The Third International Conference on Axiomatic Design 2004. p.6.

[22] Togay C, Caniaz ES, Dogru AH. Rule Based Axiomatic Design Theory Guidance for Software Development. IEEE 36th International Conference on Computer Software and Applications Workshops 2012;36:544-52.

[23] El-Haik B, Shaout A. The Employment of Axiomatic Design in Software Engineering: A Software Development Conceptual Framework. Proceedings of ACIT2011 The International Arab Conference on Information Technology 2011;1:96-107.

[24] Jamshidnezhad B. Towards a Rational Basis for User Interface Design Methods. Proceedings of ICAD2004 The Third International Conference on Axiomatic Design 2004.p.4.

[25] Helander MG, Jiao J. Coupling in Design of Human Computer Interaction. Proceedings of ICAD2002 the Second International Conference on Axiomatic Design 2002.p.5.

[26] Tseng MM , Schreyer M. Hierarchical State Decomposition for The Design of PLC Software by Applying Axiomatic Design. Proceedings of ICAD2000 the First International Conference on Axiomatic Design 2000;1:264-71.

[27] Tseng MM , Schreyer M. Model-Based Design of PLC Programs. Industrial Engineering Journal 2008.p.6.

[28] Álvares, AJ, Ferreira JC. WebTurning: Teleoperation of a CNC turning center through the Internet.Journal of Materials Processing Technology 2006;179:251–59.

[29] Álvares, AJ, Ferreira JC, Lorenzo RM. An integrated web-based CAD/CAPP/CAM system for the remote design and manufacture of feature-based cylindrical parts. Journal of intelligent manufacturing 2008;6:643-659.

[30] Álvares, AJ, Ferreira JC, Shimabukuro L, Fernandes CA. A Web-based Telemanufacturing System Oriented to the Oxi-Cutting Process. Proceedings of ENEGEP2002 The 22nd National Meeting of Production Engineering 2002.p.8.